

Maria J. de Carvalho Ferreira, filed 07/06/01, Docket # Q641  
LINEAR AND NON-LINEAR GENETIC ALGORITHMS FOR SOLVING  
PROBLEMS SUCH AS OPTIMIZATION, FUNCTION FINDING,  
PLANNING AND LOGIC SYNTHESIS



**PORTUGAL**

MINISTÉRIO DA ECONOMIA

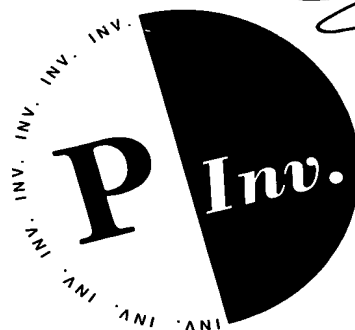
**INSTITUTO NACIONAL DA PROPRIEDADE INDUSTRIAL**

**CERTIFICADO DE PEDIDO  
DE PATENTE DE INVENÇÃO**

Certifica-se que os documentos em anexo estão conforme o original do  
pedido de patente de invenção nº. 102 508.

O pedido foi apresentado no INPI no dia 10 de Agosto de 2000.

Lisboa, 09 de Maio de 2001.



*[Signature]*  
Pelo Presidente do Conselho de Administração  
do Instituto Nacional da Propriedade Industrial



**INSTITUTO NACIONAL  
DA PROPRIEDADE INDUSTRIAL**

Campo das Cebolas - 1149-035 LISBOA - Portugal  
Telef.: +351 21 881 81 00 - Linha Azul: 808 20 06 89  
Fax: +351 21 886 00 66 - +351 21 887 53 08  
E-mail: inpi@mail.telepac.pt

Campo das Cebolas – 1149-035 LISBOA  
Telefs: 21 888 51 51 / 2 / 3  
Linha Azul: 21 888 10 78 Fax: 21 887 53 08 / 21 886 00 66  
E-mail: inpi @ mail.telepac.pt



PAT / MOD 4

FOLHA DO RESUMO

<input checked="" type="checkbox"/> PAT. INV. <input type="checkbox"/> MOD. UTI. <input type="checkbox"/> MOD. IND. <input type="checkbox"/> DES. IND. <input type="checkbox"/> TOP. SEMIC.			CLASSIFICAÇÃO INTERNACIONAL (51) 						
Nº de Objectos _____ Nº de Desenhos _____ Nº <u>102508</u> _____ (11) DATA DO PEDIDO <u>20000810</u> (22)									
REQUERENTE (71) (NOME E MORADA) <u>Maria Cândida de Carvalho Ferreira</u> <u>C. Cruz Dourada, 44</u>  CODIGO POSTAL <u>9700-570</u> <u>São Mateus</u>									
INVENTOR (ES) / AUTOR (ES) (72)  <u>Maria Cândida de Carvalho Ferreira</u>									
REIVINDICAÇÃO DE PRIORIDADE (S) (30) <table border="1"><thead><tr><th>DATA DO PEDIDO</th><th>PAÍS DE ORIGEM</th><th>Nº DO PEDIDO</th></tr></thead><tbody><tr><td> </td><td> </td><td> </td></tr></tbody></table>			DATA DO PEDIDO	PAÍS DE ORIGEM	Nº DO PEDIDO				FIGURA ( para interpretação do resumo )  
DATA DO PEDIDO	PAÍS DE ORIGEM	Nº DO PEDIDO							
EPIGRAFE (54) <u>ALGORITMOS GENÉTICOS MISTOS - LINEARES E NÃO-LINEARES - PARA RESOLVER PROBLEMAS TAIS COMO OPTIMIZAÇÃO, DESCOBERTA DE FUNÇÕES, PLANEAMENTO E SÍNTESE LÓGICA</u>									
RESUMO (max 150 palavras) (57) <p>A presente invenção é um algoritmo genético misto (linear e não-linear) capaz de aprendizagem e invenção. Uma população inicial de cromossomas lineares (entidades lineares) compostos por genes contendo as funções e argumentos dum problema, é criada e expressa sob a forma de entidades não-lineares designadas árvores de expressão. As entidades não-lineares são chamadas a actuar, produzindo resultados. Aos resultados produzidos são conferidos valores e os indivíduos respectivos (entidades lineares e respectivas entidades não-lineares) são seleccionados para reprodução de acordo com os valores auferidos. Durante o processo de reprodução, a entidade linear ou cromossoma é sujeita a um ou vários operadores, designadamente, mutação, recombinação pontual, recombinação em dois pontos, transposição, inserção e transposição génica. Desta forma indivíduos novos são criados e novamente chamados a actuar, repetindo-se o processo tantas vezes quantas as necessárias à descoberta duma solução para o problema.</p>									

## DESCRIÇÃO

**ALGORITMOS GENÉTICOS MISTOS – LINEARES E NÃO-LINEARES – PARA RESOLVER PROBLEMAS TAIS COMO OPTIMIZAÇÃO, DESCOBERTA DE FUNÇÕES, PLANEAMENTO E SÍNTESE LÓGICA**



## ESTADO DA TÉCNICA

Esta invenção está relacionada com os algoritmos genéticos e com a programação genética (inicialmente designada algoritmos genéticos não-lineares) e pode ser vista como uma síntese com propriedades emergentes dos dois sistemas.

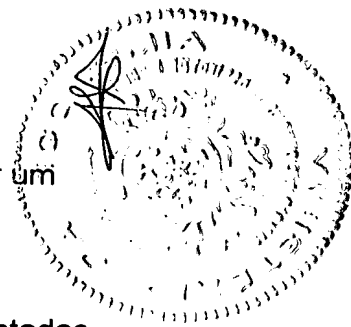
Na história da vida na terra existiram entidades de ácido ribonucleico (ARN) com actividade enzimática residual e capazes de replicação e, de facto, o ARN pode funcionar tanto como genoma como catalisador. Apesar de possível, a vida baseada em ARN foi remetida para formas de vida extremamente simples.

Por outro lado, sabe-se que o ácido desoxi-ribonucleico (ADN) é incapaz de funcionar como catalisador mas é, no entanto, a molécula ideal para armazenar e replicar a informação genética desde que existam as enzimas necessárias para catalisar as reacções. A informação genética é posteriormente expressa sob a forma de proteínas que, por sua vez, são capazes de catalisar as reacções.

Assim, na natureza existe uma divisão de tarefas entre o ADN e as proteínas: o ADN armazena a informação genética e as proteínas são a expressão dessa informação sob a forma de enzimas, proteínas estruturais, anticorpos, etc.

A programação genética desenvolvida por J. Koza é análoga a um "mundo de ARN" ou a um "mundo de proteínas", extremamente complexo para resolver problemas relativamente simples, enquanto os algoritmos genéticos desenvolvidos por J. Holland se assemelham a um "mundo" hipotético de ADN:

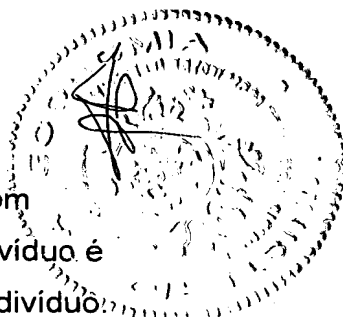
pouco complexo estruturalmente e, por conseguinte, incapaz de resolver um certo número de problemas.



As desvantagens dum sistema como os algoritmos genéticos foram apontadas por vários investigadores e para um resumo podem consultar-se os trabalhos de J. Koza. As críticas mais pertinentes dizem respeito ao alfabeto simples (normalmente 0's e 1's) e ao tamanho rígido dos cromossomas nos algoritmos genéticos que têm impedido esta técnica de ser aplicada à resolução de problemas mais sofisticados.

Com a invenção da programação genética, J. Koza resolveu em parte este problema criando entidades estruturalmente complexas de diferentes tamanhos e formas, o que permitiu a aplicação da computação evolutiva a outros problemas.

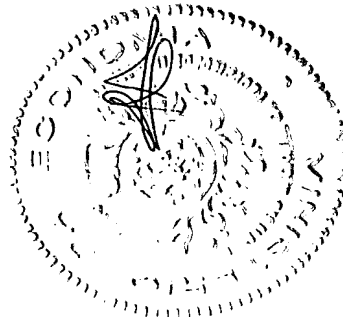
No entanto, os dois sistemas anteriores, algoritmos genéticos e programação genética, partilham um problema comum: as entidades criadas e manipuladas funcionam ao mesmo tempo como genótipo e fenótipo, o que não só limita consideravelmente o rendimento de ambas as técnicas como impede os algoritmos genéticos de serem aplicados na resolução dum vasto leque de problemas. Como já referi, o "mundo de ARN" revelou-se inviável na história da vida na terra por ser demasiado complexo para as tarefas simples que desempenhava, e não consta que tenha alguma vez existido um "mundo de ADN" pois esta molécula é estruturalmente simples, logo incapaz de actividade catalítica. Apesar de ser mais maleável, tanto estrutural como funcionalmente, a programação genética é extremamente dispendiosa em termos de recursos computacionais uma vez que a informação genética é armazenada numa estrutura extremamente complexa tornando a manipulação dessa informação praticamente incomportável. A programação genética é semelhante ao que aconteceria se para nós deixarmos descendentes fosse necessário fazer uma cópia de todas as células e constituintes do nosso corpo em vez de transmitirmos apenas o genoma no processo de reprodução. Assim, é comum à programação genética a utilização de populações relativamente grandes para resolver problemas relativamente simples (um tópico a ser analisado mais à frente) o que dificulta a sua aplicação a problemas mais complexos.



Na presente invenção os indivíduos criados são entidades complexas com propriedades emergentes, em que a informação para a criação dum indivíduo é codificada numa mensagem linear simples que constitui o genoma do indivíduo. Como na natureza, esse genoma é posteriormente expresso num ser totalmente formado com propriedades emergentes, i.e. mais complexo, estrutural e funcionalmente, que o cromossoma que o codifica.

Assim, na presente invenção existem duas entidades com estruturas e funções distintas: um genoma ou cromossoma linear que serve para armazenar a informação genética e para a transmitir às gerações futuras com modificações, e um corpo que eu designei árvore de expressão que é a expressão da informação genética codificada pelo genoma.

Desta forma, e à semelhança do que acontece na natureza, a presente invenção permite a criação de indivíduos complexos de diferentes tamanhos, formas e propriedades, apesar de serem codificados por cromossomas lineares de tamanho fixo. Assim, a manipulação da informação genética, fundamental para que haja evolução no sistema, logo fundamental para resolver problemas, é extremamente simples e tão leve, em termos de recursos computacionais, como a manipulação dos cromossomas dos algoritmos genéticos. As modificações efectuadas nos descendentes criados são testadas quando o genoma do indivíduo é expresso e, como na natureza, se a variação conferir vantagens ao descendente, esse indivíduo terá maior probabilidade de sobreviver e deixará mais descendentes; o contrário acontecerá se a variação for desvantajosa: esse indivíduo deixará menos descendentes ou desaparecerá da população.



## REFERÊNCIAS CITADAS

### Patentes dos Estados Unidos da América

4697242. *Adaptive Computing System Capable of Learning and Discovery*. September 29, 1987. Holland, J. H., and Burks, A. W.

4935877. *Non-Linear Genetic Algorithms for Solving Problems*. June 19, 1990. Koza, J. R.

### Outras Publicações

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. Ann Arbor, MI: University of Michigan Press.

Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press.

O'Reilly, U-M, and F. Oppacher (1996). *A comparative analysis of Genetic Programming*. Chapter 2 of *Advances in Genetic Programming 2*, ed. P. J. Angeline and K. E. Kinnear, MIT Press.

## DESCRIÇÃO DAS FIGURAS



- A Fig. 1 ilustra a representação em árvore duma expressão matemática convencional, duma S-expressão em LISP e dum cromossoma da invenção presente.
- A Fig. 2 é um fluxograma do processo da invenção presente
- A Fig. 3 ilustra a organização estrutural do cromossoma
- A Fig. 4 é a expressão dum cromossoma na árvore de expressão
- A Fig. 5 é a ilustração do mecanismo de mutação.
- A Fig. 6 é a ilustração do mecanismo de transposição.
- A Fig. 7 é a ilustração do mecanismo de inserção.
- A Fig. 8 é a ilustração do mecanismo de transposição génica.
- A Fig. 9 é a ilustração do mecanismo de recombinação pontual.
- A Fig. 10a e 10b é a ilustração do mecanismo de recombinação entre dois pontos.
- A Fig. 11 é uma população inicial de 30 cromossomas aleatórios criada para resolver o problema da regressão simbólica.
- A Fig. 12 é o melhor indivíduo da população inicial do problema da regressão simbólica.
- A Fig. 13 é a solução correcta para o problema da regressão simbólica.
- A Fig. 14 é uma população inicial de 30 cromossomas aleatórios criada para resolver o problema do empilhamento de blocos e as respectivas aptidões para um caso concreto de estados iniciais.
- A Fig. 15 é o primeiro módulo útil descoberto no problema do empilhamento de blocos que esvazia todas as gavetas.
- A Fig. 16 é o segundo módulo útil descoberto no problema do empilhamento de blocos que enche parcialmente todas as gavetas.
- A Fig. 17 é solução correcta do problema do empilhamento de blocos que enche completamente todas as gavetas.
- A Fig. 18 é o melhor indivíduo criado na população inicial no problema do multiplexer de 6 bits.
- A Fig. 19 é o primeiro módulo descoberto no problema do multiplexer de 6 bits que resolve correctamente um endereço.

A Fig. 20 é o segundo módulo descoberto no problema do multiplexer de 6 bits que resolve correctamente dois endereços.

A Fig. 21 é o terceiro módulo descoberto no problema do multiplexer de 6 bits que resolve correctamente três endereços.

A Fig. 22 é a solução do problema do multiplexer de 6 bits que resolve correctamente os quatro endereços.

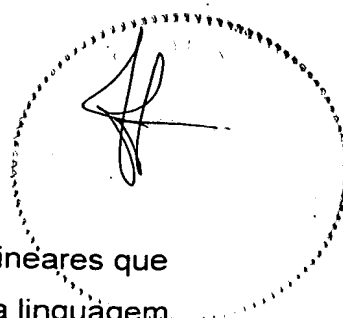
#### DESCRIÇÃO DAS TABELAS

A Tab. 1 é a comparação entre a invenção presente e a programação genética no problema da regressão simbólica.

A Tab. 2 é a comparação entre a invenção presente e a programação genética no problema do empilhamento de blocos.



## DESCRIÇÃO DETALHADA DA INVENÇÃO



A programação genética inventada por J. Koza utiliza entidades não-lineares que são diagramas em forma de árvore que representam S-expressões na linguagem de programação LISP. A Fig. 1 representa uma expressão matemática convencional 101; a sua S-expressão em LISP 102; o respectivo diagrama em forma de árvore 103; e a sua expressão num cromossoma da invenção presente 104. O símbolo Q 105 no cromossoma 104 representa a função raiz quadrada.

A programação genética cria populações iniciais de árvores como a representada na Fig. 1, e são estas entidades não-lineares que posteriormente são reproduzidas, recombinadas, permutadas ou, raramente, mutadas (operadores genéticos usados pela programação genética). No entanto, estas manipulações genéticas são extremamente complicadas e problemáticas neste sistema, pois a substituição de uma função por um argumento ou vice versa, ou, por exemplo, a substituição de uma função de dois argumentos por outra de um argumento, como a substituição de '\*' por 'sqrt', torna a árvore e a S-expressão correspondente inválidas. O mesmo problema se coloca durante a permutação onde certos nódulos são permutados. Por isso a programação genética usa quase exclusivamente a recombinação uma vez que a linguagem LISP facilita este tipo de operação. Desta forma, não é fácil para a programação genética a introdução de variedade na população, que é a matéria prima da evolução. Uma forma de resolver o problema, consiste na criação de populações iniciais gigantescas, em que todas as entidades são diferentes de modo a garantir suficiente diversidade logo entre as entidades da população inicial, de forma a ser possível, com uma certa probabilidade, descobrir uma solução para o problema só por recombinação do material gerado na população inicial. Este é um dos motivos pelos quais a programação genética é extremamente dispendiosa e pouco eficiente.

A invenção presente, com a criação de um genótipo e de um fenótipo, permite o uso sem restrições de vários operadores genéticos, alguns dos quais fazem parte integrante desta invenção, designadamente a mutação, transposição,

inserção, transposição génica, recombinação pontual e recombinação em dois pontos.

A invenção presente partilha com a programação genética uma forma idêntica de representação dos problemas em árvore, mas as expressões que codificam as árvores da invenção presente não são S-expressões em LISP: são expressões com uma organização estrutural e funcional única que foram desenvolvidas por mim e que são peças fundamentais da invenção presente. Estas expressões constituem, na verdade, o genoma dos indivíduos desta invenção.

Nesta descrição será apresentado um método geral e exemplos específicos da invenção presente. Primeiro, o processo é descrito detalhadamente; segundo, são dados três exemplos da sua actuação. Os três exemplos escolhidos ilustram a actuação da invenção presente num problema de regressão simbólica, num problema de planeamento (empilhamento de blocos) e num problema de desenho de circuitos electrónicos (multiplexer de 6 bits).

A Fig. 2 representa um fluxograma do processo da invenção presente. O processo 201 começa com o passo 202 Criar Cromossomas da População Inicial, em que se cria aleatoriamente um número determinado de cromossomas. O ciclo repetitivo fundamental do processo começa no passo 203 Expressar Cromossomas, onde a linguagem dos cromossomas é expressa na linguagem das árvores de expressão. Com o passo 204 Executar Cada Programa, cada programa é chamado a actuar, sendo o resultado desta actuação calculado no passo seguinte 205 Calcular Aptidão. Nesta fase verifica-se se já se chegou a uma solução satisfatória para o problema ou se qualquer outro critério de terminação do processo foi satisfeito. Se a condição de terminação for satisfeita, o processo termina em 206 Fim. Caso contrário, o processo continua a repetir-se.

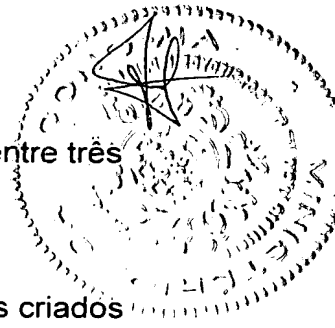
No passo 207 Guardar Melhor Programa, conserva-se o programa com maior aptidão que é copiado e passado sem modificações para a geração seguinte. No passo seguinte 208 Seleccionar Programas de Acordo com Aptidão, os programas são seleccionados proporcionalmente à aptidão respectiva, o que

significa que os indivíduos mais aptos têm maior probabilidade de deixar mais descendentes. No entanto, o tipo de selecção usado na presente invenção envolve um factor sorte e às vezes os melhores indivíduos podem morrer sem deixar descendentes. Este tipo de selecção assemelha-se ao que acontece na natureza e é normalmente usado em vários sistemas de computação evolutiva, designadamente os algoritmos genéticos. Por este motivo, a invenção presente pratica aquilo que se costuma designar de elitismo (passo 207) em computação evolutiva, e conserva o melhor indivíduo de cada geração que é replicado directamente para a geração seguinte (passo 217).

Os sete passos seguintes fazem parte do processo de reprodução 209. No passo 210 Replicar Programas, os cromossomas são replicados para serem transmitidos à geração seguinte. A replicação não introduz variação na população: se a reprodução consistisse somente na replicação, teríamos os mesmos indivíduos a perpetuarem-se indefinidamente e a população a tornar-se cada vez menos diversa, pois alguns deles não teriam sorte no processo de selecção. A variação surge somente com a aplicação dos operadores genéticos: passo 211 Aplicar Mutação, passo 212 Aplicar Transposição, passo 213 Aplicar Inserção, passo 214 Aplicar Transposição Génica, passo 215 Aplicar Recombinação Pontual e passo 216 Aplicar Recombinação em Dois Pontos. Cada um destes passos ocorre sequencialmente como indicado na Fig. 2, sendo todos os cromossomas escolhidos aleatoriamente e sujeitos a cada um dos operadores genéticos. Desta forma, na invenção presente é pouco provável a criação de descendentes exactamente iguais aos parentes, sendo, por isso, extremamente criativa e extremamente eficiente na resolução de problemas.

Assim, na invenção presente é utilizado um total de seis operadores genéticos criadores de variação. Normalmente, os operadores são escolhidos consoante a natureza do problema e, por vezes, utiliza-se uma combinação dos diferentes operadores para um determinado problema. O mais comum é, no entanto, a utilização do leque inteiro de operadores. Este leque de operadores é mais que suficiente para criar a variação necessária para a evolução, no entanto, outros operadores podem ser facilmente criados como a transposição

intercromossômica, recombinação em múltiplos pontos, recombinação entre três ou mais parentes, deleções, etc.



Depois de completo o processo de reprodução 209, os programas novos criados constituem os indivíduos da geração seguinte que são adicionados no passo 217 Adicionar Programas Novos à Geração Seguinte, repetindo-se novamente o ciclo com a volta ao passo 203 Expressar Cromossomas.

De seguida faz-se uma análise mais pormenorizada dos passos mais importantes do processo 201.

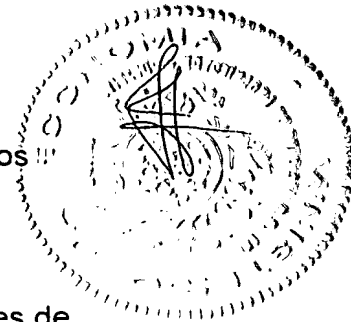
Durante a criação da população inicial, o genoma (cromossomas) de todos os indivíduos é criado aleatoriamente usando os símbolos das funções e terminais (argumentos) escolhidos para resolver o problema em questão. Os cromossomas são entidades lineares de tamanho fixo, compostos por um ou mais genes. Os genes estão organizados estruturalmente numa cabeça e numa cauda. A cabeça contém símbolos que representam tanto funções como terminais, enquanto a cauda contém somente terminais, e funciona como uma espécie de reservatório de terminais. Para cada problema específico, o tamanho da cabeça e da cauda é calculado de forma a que todos os indivíduos gerados sejam programas estrutural e funcionalmente correctos. Assim, após a escolha do tamanho da cabeça, o tamanho da cauda é calculado pela fórmula:

$$t = h \cdot (n - 1) + 1,$$

onde  $t$  é o tamanho da cauda;  $h$  é o tamanho da cabeça; e  $n$  é o número de argumentos da função com maior número de argumentos utilizada. Por exemplo, se o conjunto de funções utilizadas for  $\{*, /, -, +, Q\}$  (multiplicação, divisão, subtracção, adição e raiz quadrada, respectivamente, tomando as quatro primeiras 2 argumentos e a última 1 argumento), o  $n$  é igual a 2.

A Fig. 3 representa um cromossoma 301 de 27 de tamanho, composto por 3 genes (302, 303, 304) em que a cabeça (305, 307, 309) é igual a 4 e a cauda (306, 308, 310) é igual a 5. Os símbolos  $\{*, /, -, +, Q\}$  representam as funções

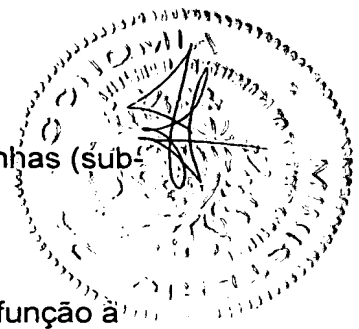
escolhidas e os símbolos  $\{a, b\}$  representam os argumentos simbólicos (terminais) escolhidos para resolver o problema em questão.



Os cromossomas são posteriormente expressos sob a forma de árvores de expressão que são as entidades que executam uma determinada tarefa produzindo um resultado. Esse resultado é posteriormente associado a um valor que determina a selecção.

A Fig. 4 mostra de que forma é feita a expressão dum cromossomas 401 na respectiva árvore de expressão 429. Primeiro, os genes individuais (402, 403, 404) são expressos sob a forma de sub-árvores 405, sendo a expressão directa e extremamente simples: o primeiro elemento na raiz de uma sub-árvore é o símbolo que está no início do gene respectivo ('Q' 409 no caso do gene 1 402; '\*' 413 no caso do gene 2 403; '\*' 422 no caso do gene 3 404); na segunda linha da sub-árvore, colocam-se os argumentos da função colocada na primeira linha e que ocupam, no respectivo gene, as posições imediatamente a seguir à primeira posição (para o gene 1 402, '+' 410 é o argumento de 'Q' 409; para o gene 2 403, '-' 414 e '-' 415 são os argumentos de '\*' 413 enquanto 'a' 418 e 'b' 419 são os argumentos de '-' 415; para o gene 3 404, '/' 423 e 'b' 424 são os argumentos de '\*' 422); na terceira linha colocam-se os argumentos das funções que aparecem na segunda linha e que ocupam, nos respectivos genes, as posições imediatamente a seguir aos símbolos utilizados (para o gene 1 402, 'a' 411 e 'a' 412 são os argumentos de '+' 410; para o gene 2 403, '-' 416 e 'b' 417 são os argumentos de '-' 414; para o gene 3 404, '-' 425 e 'b' 426 são os argumentos de '/' 423); na quarta linha colocam-se os argumentos das funções que aparecem na terceira linha e que ocupam, nos respectivos genes, as posições imediatamente a seguir aos símbolos previamente utilizados (para o gene 1 402, a expressão está completa, pois a terceira linha é constituída somente por terminais; para o gene 2 403, 'b' 420 e 'a' 421 são os argumentos de '-' 416; para o gene 3 404, 'b' 427 e 'a' 428 são os argumentos de '-' 425). Este processo repete-se para cada gene até se atingir uma linha de fundo composta exclusivamente por terminais. No caso concreto da Fig. 4, o primeiro gene 402 é expresso em 3 linhas (sub-árvore 406), acabando no ponto de terminação 432; o gene 2 403 é expresso em 4 linhas (sub-árvore 407), coincidindo o ponto de

terminação 433 com o fim do gene; o gene 3 404 é expresso em 4 linhas (sub-  
árvore 408), acabando no ponto de terminação 434.

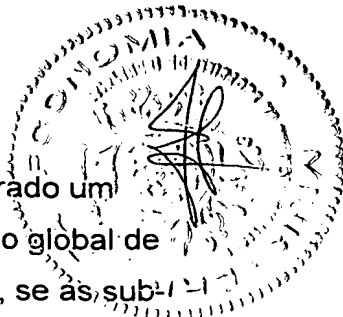


As sub-árvores (406, 407, 408) são posteriormente ligadas por uma função à  
escolha, sendo, no caso da Fig. 4, ligados pela adição (430 e 431). Após a  
ligação das sub-árvores obtém-se finalmente a árvore de expressão 429 que  
produz o resultado que é associado posteriormente a um valor que determina a  
selecção.

Repare-se que as funções de ligação (430 e 431) escolhidas para ligar as sub-  
árvores (406, 407, 408) não são codificadas pelo genoma. Esta característica da  
invenção presente assemelha-se ao processamento pós-traducional das  
proteínas que existe na natureza, como por exemplo a ligação de diferentes sub-  
unidades proteicas codificadas por diferentes genes que se ligam formando uma  
proteína gigante composta por múltiplas sub-unidades.

Assim, as árvores de expressão da invenção presente assumem, como as  
árvores da programação genética, diferentes formas e tamanhos apesar de  
serem codificadas por um genoma linear de tamanho fixo. É igualmente  
importante salientar o uso de genes pela invenção presente, pois a sua  
existência permite de facto o aparecimento de blocos simples que se combinam  
para formar estruturas cada vez mais complexas, tornando a invenção presente  
num sistema de invenção verdadeiramente hierárquico. A programação genética  
também é conhecida por algoritmos genéticos hierárquicos, mas subsistem  
várias dúvidas quanto ao seu funcionamento hierárquico. O facto dos programas  
da programação genética consistirem numa só árvore limita grandemente a  
descoberta de módulos mais simples e a sua utilização posterior em árvores  
mais complexas. De facto, se na presente invenção os cromossomas forem  
constituídos por um único gene, o sistema descobre módulos simples que  
aproveita mais tarde para criar indivíduos mais complexos, mas fá-lo de forma  
muito mais ineficiente do que faria num sistema multigénico.

O sistema multigénico da invenção presente permite também o aparecimento de  
genes neutros (genes que não fazem nada), que são fundamentais para a

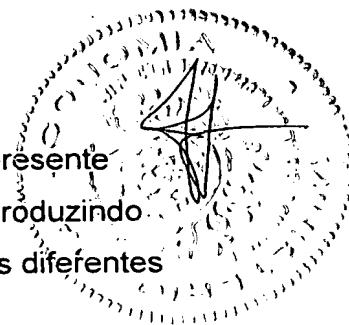


evolução do sistema. Neste caso, um gene neutro pode ser considerado um gene cuja sub-árvore devolva um valor que não influencia o resultado global de todas as sub-árvores codificadas por um cromossoma. Por exemplo, se as sub-árvores forem ligadas pela adição, um gene neutro codifica para uma sub-árvore que devolva zero; num problema booleano em que as sub-árvores sejam ligadas por OR, um gene neutro codifica para uma sub-árvore que devolva zero. Estes genes são regiões óptimas para a acumulação de mutações, pois podem a qualquer momento ser modificados e transformarem-se num gene com expressão.

Como se mostra na Fig. 4, os diferentes genes dum cromossoma são expressos em sub-árvores de diferentes formas e tamanhos e, na maior parte dos casos, para a expressão completa dum gene não são necessários todos os símbolos do gene. Por exemplo, o gene 1 402 da Fig. 4 com 9 de tamanho, é expresso numa sub-árvore 406 de 4 nós (409, 410, 411, 412). Estas regiões não funcionais que não são utilizadas na expressão dum gene, são pontos ideais para o acumular de mutações neutras, pois qualquer que seja a mutação que ocorra a jusante do local de terminação da expressão (432, 434) não tem efeito no produto de expressão do gene e, por conseguinte, não estão sujeitos a pressões de selecção. No entanto, e à semelhança do que acontece na natureza, estas regiões desempenham um papel fundamental na evolução, pois aqueles elementos podem a qualquer momento ser activados por um operador genético e ser integrados na região funcional de um gene.

É importante salientar que com o sistema presente de representação, os cromossomas constituem em si uma linguagem nova de programação, simples e intuitiva, que pode ser utilizada para programar qualquer computador. As operações que podem ser executadas neste sistema correspondem aos operadores matemáticos e lógicos utilizados por qualquer linguagem de programação convencional e outros operadores não-convencionais como, por exemplo, as acções 'A' (Fazer Até Verdadeiro), 'R' (Remover Letra da Gaveta) e 'C' (Carregar Letra para Gaveta) criadas para resolver o problema do empilhamento de blocos apresentado mais à frente nesta descrição.

Igualmente importante, é o facto de os cromossomas da invenção presente poderem ser modificados por qualquer dos operadores genéticos, produzindo sempre indivíduos correctos. De seguida será analisado o efeito dos diferentes operadores genéticos implementados na invenção presente.



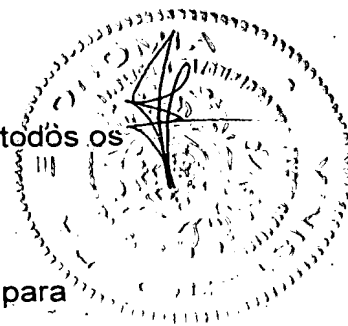
Após a selecção proporcional à aptidão, os indivíduos são reproduzidos. Como na natureza, durante o processo de reprodução os genomas dos indivíduos sofrem várias modificações que são o resultado da mutação, transposição, inserção, recombinação e outros operadores genéticos, gerando desta forma a variedade genética fundamental para a evolução. Na invenção presente, os cromossomas são sujeitos a um ou vários operadores genéticos à escolha, produzindo-se assim a variação genética necessária à descoberta de soluções.

A mutação caracteriza-se por uma modificação pontual num ou mais símbolos que constituem o genoma. No caso da invenção presente, os símbolos da cabeça dos genes podem ser substituídos por qualquer dos símbolos (funções e terminais) escolhidos para resolver o problema em questão, enquanto que na cauda, os símbolos podem ser substituídos por qualquer dos terminais escolhidos para resolver o problema. Desta forma, a organização estrutural do cromossoma não é destruída e todos os indivíduos novos criados por mutação são indivíduos estrutural e funcionalmente correctos.

Na Fig. 5 está representado o mecanismo de mutação. Suponhamos que, no cromossoma 501, a função '-' 502 era substituída por 'Q' 506, que a função '/' 503 era substituída pela função 'Q' 507, e que a função 'Q' 504 era substituída por 'b' 508. A comparação das árvores antes (509) e após (510) a mutação, mostra quão profundo pode ser o efeito da mutação neste sistema. Repare-se que a substituição do '/' 503 pelo 'Q' 507 é um exemplo duma mutação neutra. É evidente que o exemplo escolhido é um pouco extremo, pois na realidade escolhem-se taxas de mutação de forma a criar-se a variação ideal. Normalmente utiliza-se uma taxa de mutação que é equivalente a duas mutações por cromossoma, sendo os cromossomas normalmente maiores do que os apresentados na Fig. 5. Repare-se, no entanto, que não há qualquer tipo de



restrição nem quanto ao número nem quanto ao tipo de mutação: em todos os casos os indivíduos novos criados são funcionais.



Depois da mutação, alguns indivíduos são aleatoriamente escolhidos para sofrerem transposição, sendo, por cada cromossoma, escolhido aleatoriamente um único gene para sofrer transposição. A transposição intracromossômica de elementos de transposição (transposões) está ilustrada na Fig. 6. Os transposões (602, 604) têm diferentes tamanhos e são escolhidos aleatoriamente entre os elementos da cabeça, tendo sempre na primeira posição uma função. Com este tipo de transposição, os transposões saltam para o início do gene. Para ilustrar este operador, foram escolhidos cromossomas de 42 de tamanho (601, 603), compostos por dois genes (605 e 606 no cromossoma 601, 607 e 608 no cromossoma 603) com 21 de tamanho cada.

Suponhamos que era escolhido aleatoriamente o transposão '+bb' 602 de 3 de tamanho no gene 2 606. Este operador efectua uma cópia 604 do transposão 602 para o início do gene 2' 608, deslocando toda a cabeça para acomodar a cópia 604 do transposão, ao mesmo tempo que os 3 últimos símbolos da cabeça vão desaparecendo na fronteira com a cauda, ficando a cauda inalterada neste processo. Deste modo garante-se a manutenção da organização estrutural do cromossoma. Repare-se também que os restantes genes (gene 1' 607) não são alterados no processo.

Este tipo de transposição permite a cópia de pequenos blocos e faculta a sua propagação na população. Como a mutação descrita acima, a transposição tem um grande poder transformador e é excelente para criar variedade. Repare-se que as sub-árvores afectadas (sub-árvore 609 antes da transposição, sub-árvore 610 após a transposição) por este operador são modificadas de forma radical, pois é a própria raiz que é afectada. Este tipo de operadores impedem que as populações fiquem estagnadas em mínimos locais, encontrando fácil e rapidamente soluções para os problemas.

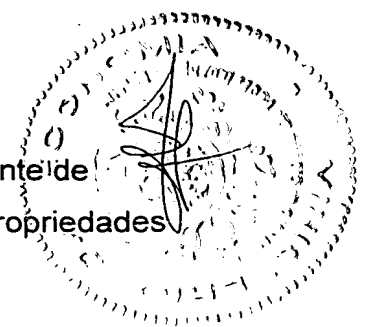
Após a transposição, alguns cromossomas escolhidos aleatoriamente são sujeitos à inserção intracromossômica. Por cada cromossoma sujeito a inserção,

um único gene é escolhido aleatoriamente para sofrer inserção. A inserção é um caso mais geral de transposição em que elementos de inserção de diferentes tamanhos são escolhidos aleatoriamente e inseridos na cabeça, com excepção da raiz do gene (a transposição descrita acima tem o mesmo efeito) e de algumas posições (tantas quanto o tamanho do elemento de inserção) no extremo direito da cabeça. Os elementos de inserção são escolhidos aleatoriamente ao longo de todo cromossoma, podendo ter qualquer sequência (qualquer posição pode ser ocupada tanto por funções como por terminais).

A Fig. 7 ilustra a inserção de um elemento de inserção de 3 de tamanho. Neste caso, foi escolhido o elemento de inserção 'bba' 701 e inserido no sítio de inserção 702 (sexta ligação). Este operador efectua uma cópia 703 do elemento de inserção 701, inserindo-o no sítio de inserção 704, ficando o cromossoma inalterado a montante do sítio de inserção 704; a jusante da cópia 703 do elemento de inserção, os restantes elementos da cabeça são deslocados para acomodar a cópia 703 do elemento de inserção, ao mesmo tempo que os últimos elementos da cauda (tantos quantos os símbolos do elemento de inserção) são removidos, garantindo, assim, a integridade estrutural do cromossoma.

Repare-se que quando se dá a expressão do indivíduo novo criado 708, os símbolos do elemento de inserção (705, 706, 707) distribuem-se de forma quase imprevisível, ficando a maior parte das vezes separados no espaço e integrando módulos funcionais distintos. Isto assemelha-se ao que acontece na natureza, em que o enrolamento das proteínas na sua estrutura tridimensional aproxima aminoácidos que estão codificados no ADN em sítios afastados. Assim, como a natureza, a invenção pressente trabalha de forma cega: as modificações efectuadas nos indivíduos têm pouco a ver com as modificações que um matemático faria, mas que, no entanto, funcionam. Atente-se que a programação genética, com os operadores que usa directamente sobre as árvores, opera mais à semelhança da forma lógica e calculada dum matemático do que da forma cega da natureza, recombina e permutando módulos matematicamente concisos.

Como os operadores acima descritos, a inserção é uma excelente fonte de variação genética, criando indivíduos novos capazes de expressar propriedades diferentes das dos seus parentes.



Após a inserção, alguns cromossomas são escolhidos aleatoriamente para serem modificados por transposição génica. A transposição génica intracromossómica é um caso particular de transposição em que um gene que não o primeiro é escolhido aleatoriamente e transposto na sua totalidade para o início do cromossoma. Durante o processo, este gene é removido da posição que ocupava no cromossoma e integralmente copiado no início do cromossoma.

A Fig. 8 ilustra o mecanismo de transposição génica. Neste caso, o gene 2 802 é transposto para o início do cromossoma. Como resultado, o gene 2 (802) passa a ser o primeiro (804), o gene 1 (801) passa a ser o segundo (805) e o gene 3 (803) continua na terceira posição (806).

Repare-se que no caso apresentado na Fig. 8, em que as sub-árvores codificadas por cada um dos genes são ligadas pela adição, o indivíduo resultante é funcionalmente equivalente ao parente, pois a adição goza da propriedade comutativa. No entanto, no caso de sub-árvores ligadas por outras funções como, por exemplo, a função booleana  $\text{if}(x,y,z)$  (se o  $x$  for '1' devolve o valor de  $y$ , se for '0' devolve o valor de  $z$ ), o indivíduo novo formado no processo de transposição génica já não é equivalente ao seu parente.

No entanto, o poder transformador da transposição génica, reside na conjugação deste operador com outros operadores, particularmente a recombinação em que os cromossomas de dois parentes trocam partes entre si. Assim, se tivermos dois cromossomas funcionalmente idênticos ou com um gene comum em posições distintas e se os sujeitarmos a recombinação, pode nesse processo criar-se um indivíduo novo com um gene duplicado. Sabe-se que a duplicação de genes desempenha um papel importante na biologia e na evolução e, de facto, na invenção presente é comum o aparecimento de genes duplicados no processo de descoberta de soluções.

Após a transposição génica, alguns cromossomas são escolhidos aleatoriamente aos pares para sofrerem recombinação pontual. Durante a recombinação pontual os dois parentes trocam partes dos respectivos genomas entre si.



A Fig. 9 ilustra a recombinação pontual entre dois cromossomas (901, 902) de 18 de tamanho, compostos por dois genes. O ponto de recombinação (903, 904) (terceira ligação), é escolhido aleatoriamente e os cromossomas emparelhados são rompidos pelo ponto de recombinação (903, 904), trocando entre si os fragmentos a jusante do ponto de recombinação. Com este tipo de recombinação, os parentes (901, 902) geram dois indivíduos novos (905, 906) que diferem, na maior parte dos casos, tanto entre si como dos parentes. Repare-se que, neste caso, as árvores de expressão dos parentes (907, 908) e as árvores de expressão dos indivíduos novos (909, 910) diferem todas entre si.

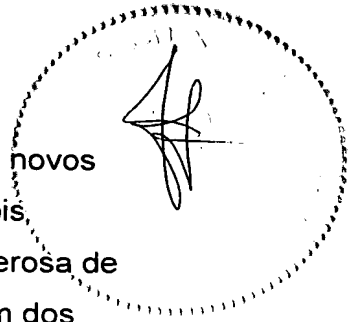
Assim, a recombinação pontual, como os operadores acima mencionados, constitui um fonte importante de variação genética, sendo, depois da mutação, o operador genético mais escolhido na invenção presente.

Após a recombinação pontual, alguns cromossomas são escolhidos aleatoriamente e sujeitos a recombinação entre dois pontos.

O tipo de recombinação entre dois pontos usado na invenção presente, é feito de forma a permitir a troca de genes completos que ocupam a mesma posição nos cromossomas dos parentes a recombinar. Assim, com este tipo de recombinação, os cromossomas dos parentes são emparelhados e o gene escolhido aleatoriamente é trocado entre os dois parentes.

As Fig. 10a e 10b ilustram a recombinação entre dois pontos entre dois cromossomas (1001, 1002) compostos por três genes. Neste caso, o gene 2 (1003, 1004) foi escolhido para ser trocado entre os dois parentes, quebrando-se os cromossomas pelas ligações que limitam o gene. Como resultado, formam-se dois indivíduos novos (1005, 1006) com genomas contendo genes de ambos os parentes.

Repare-se que, como no caso da recombinação pontual, os indivíduos novos formados diferem, a maior parte das vezes, tanto entre si como dos dois parentes. A recombinação entre dois pontos é também uma fonte poderosa de variação genética, sendo, juntamente com a recombinação pontual, um dos operadores mais usados na invenção presente.



É importante salientar que na invenção presente, o número e o tipo de operadores genéticos é escolhido pelo investigador, sendo, a maior parte das vezes, utilizada uma combinação de dois ou mais operadores para criar variação genética na população e, desta forma, garantir a descoberta duma solução para o problema em questão. No entanto, e ao contrário do que acontece na programação genética, os cromossomas escolhidos para serem modificados por qualquer dos operadores, são todos escolhidos aleatoriamente, podendo, por conseguinte, o mesmo cromossoma ser escolhido para ser modificado por mais do que um operador, acumulando várias transformações, o que torna a invenção presente extremamente inventiva no que respeita à descoberta de soluções para os problemas. Esta é de facto uma das razões que permitem que a invenção presente descubra soluções utilizando, para os mesmos problemas, populações que são, a maior parte das vezes, mais de uma ordem de grandeza inferiores às utilizadas pela programação genética (por exemplo, para o problema da regressão simbólica, a programação genética usa populações de 500 indivíduos, enquanto a invenção presente utiliza populações de somente 30 indivíduos; vejam-se os exemplos apresentados mais adiante).

Outra diferença entre a invenção presente e a programação genética, consiste no leque de operadores genéticos usados pelos dois sistemas e na forma como são utilizados. Ao contrário da invenção presente, a programação genética utiliza quase exclusivamente a recombinação pontual, sendo a mutação usada muito raramente. Além do mais, na programação genética os cromossomas ou são seleccionados para recombinar ou para mutar, nunca sendo sujeitos a mais do que um operador no mesmo ciclo reprodutivo. Estas são mais algumas das razões que obrigam à utilização de populações gigantescas pela programação genética, já que a diversidade tem que estar presente logo entre as entidades da população inicial (de facto, a programação genética gasta imensos recursos

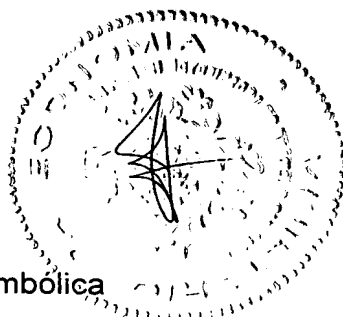
computacionais para garantir que todas as entidades não-lineares da população inicial sejam diferentes umas das outras), sendo só por recombinação do material presente na população inicial que a programação genética descobre soluções. Só com a utilização de populações gigantescas consegue a programação genética garantir, com uma certa probabilidade, que todos os elementos necessários para a descoberta de uma solução estejam presentes logo na população inicial.

Por outro lado, na programação genética, as entidades sujeitas ao operador seleccionado são cuidadosamente escolhidas, o que torna a programação genética extremamente dispendiosa em termos de recursos computacionais, pois é obrigada a manter o registo da aptidão de todas as entidades da população.

Na invenção presente, com a recombinação em dois pontos completa-se o processo de reprodução e os cromossomas novos formados constituem o genoma dos indivíduos da geração seguinte. Estes indivíduos são, por sua vez, sujeitos ao mesmo processo de maturação (expressão dos genomas nas árvores respectivas), crescimento (confrontar o ambiente de selecção) e reprodução.

De seguida apresentam-se os três exemplos escolhidos para ilustrar a invenção presente: regressão simbólica, empilhamento de blocos e multiplexer de 6 bits.

## REGRESSÃO SIMBÓLICA OU DESCOBERTA DE FUNÇÕES



Este problema de regressão envolve a descoberta duma expressão simbólica que satisfaça um conjunto de valores, designados casos de aptidão. Os casos de aptidão constituem o ambiente de selecção onde decorrerá a adaptação dos indivíduos.

Para ilustrar este tipo de problemas foi escolhida a função:

$$y = a^4 + a^3 + a^2 + a$$

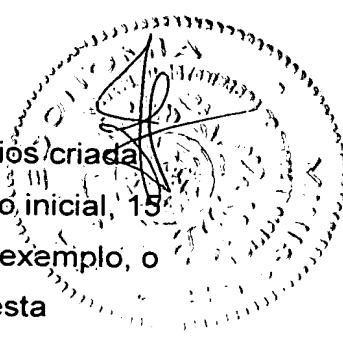
por ser uma função duma certa complexidade e também por a mesma função ter sido utilizada por J. Koza, permitindo, por conseguinte, a comparação da invenção presente com a programação genética.

Para problemas deste tipo, a invenção presente utiliza habitualmente um conjunto de 10 casos de aptidão (resultados experimentais), sendo o objectivo descobrir uma expressão que satisfaça estes valores dentro de 0,01 do valor correcto.

Para resolver este problema, foram escolhidas as funções  $\{+, -, *, /\}$  e o terminal  $\{a\}$ . Com estes símbolos foi criada uma população inicial de 30 cromossomas compostos de 4 genes de 11 de tamanho. Os cromossomas foram posteriormente expressos e as suas aptidões determinadas para o conjunto de casos de aptidão. Neste caso, para cada caso de aptidão, a aptidão foi calculada pela fórmula:

$$f = M - |E|$$

onde  $M$  é a margem de selecção, e  $E$  é o valor do erro absoluto entre o valor produzido pelo indivíduo e o valor correcto. A margem de selecção é escolhida para cada problema em questão, sendo neste caso igual a 100. Então, se  $E$  for menor ou igual a 0,01,  $f = 100$ . Assim, se os 10 casos de aptidão forem calculados de forma exacta ou dentro do erro absoluto (0,01) a aptidão máxima ( $f_{max}$ ) é igual a 1000.



A Fig. 11 mostra uma população inicial 1101 de cromossomas aleatórios criada numa experiência. Repare-se que dos indivíduos criados na população inicial, 15 deles revelaram-se aptos para resolver parcialmente o problema. Por exemplo, o melhor indivíduo desta geração tem  $f = 97.6903$ , o que significa que nesta população não existem indivíduos capazes de resolver um único caso de aptidão dentro do erro absoluto (0,01). Mas estes indivíduos toscos conseguem gerar descendentes 100% aptos na resolução do problema, como se verá de seguida.

A Fig. 12 mostra o cromossoma do melhor indivíduo 1201 criado na população inicial, e a árvore de expressão respectiva 1202 juntamente com a expressão matemática correspondente 1203. O resultado devolvido por cada sub-árvore está contido entre parêntesis na expressão matemática 1203 para mais facilmente permitir a análise. Repare-se que, após a simplificação, somente um dos termos ( $a^4$ ) da expressão matemática 1203 coincide com a função alvo.

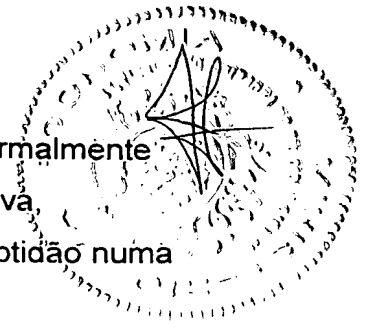
Os indivíduos aptos da população inicial são posteriormente seleccionados de acordo com a aptidão, acabando por se reproduzirem criando os indivíduos novos da geração seguinte. Este processo é repetido num total de 50 gerações.

Na Fig. 13 está representado o descendente (1301) dos indivíduos criados na população inicial capaz de resolver este problema correctamente e, portanto, com aptidão máxima 1000. Este indivíduo foi criado ao fim de 8 gerações, e a sua árvore de expressão 1302 traduz-se numa expressão matemática 1303 equivalente à função alvo.

O cálculo da taxa de sucesso na resolução deste problema foi determinado avaliando 100 experiências idênticas à acima descrita (o conjunto das 100 experiências designa-se corrida), tendo atingido o valor máximo de 1, o que significa que em todas as experiências foi encontrada uma solução correcta para o problema. Se compararmos o rendimento da invenção presente com o rendimento da programação genética na resolução deste problema, verifica-se que a invenção presente é 374 vezes superior à programação genética.



A medida utilizada para comparar os dois sistemas, é a medida normalmente utilizada para comparar diferentes sistemas de computação evolutiva, consistindo no cálculo do número de vezes que é determinada a aptidão numa corrida com um número razoável de experiências.



Na Tab. 1 é feita a comparação entre a invenção presente e a programação genética para uma corrida de 100 experiências idênticas.

O número de experiências ( $R_z$ ) necessário para descobrir uma solução correcta ao fim de  $G$  gerações, com uma probabilidade ( $z$ ) de 0,99, é calculado pela fórmula:

$$R_z = \frac{\log(1-z)}{\log(1-P_s)}, \text{ sendo } P_s \neq 1$$

onde  $z$  é a probabilidade e  $P_s$  é a taxa de sucesso. Se  $P_s = 1$ , então  $R_z = 1$ .

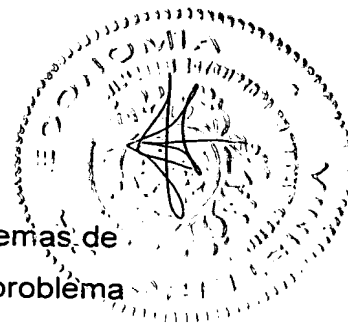
O número de cálculos da aptidão necessário para encontrar uma solução para o problema com uma probabilidade ( $z$ ) de 0,99 ( $F_z$ ) é calculado pela fórmula:

$$F_z = G \cdot P \cdot C \cdot R_z$$

onde  $G$  é o número de gerações;  $P$  é o tamanho da população; e  $C$  é o número de casos de aptidão.

Assim, a comparação dos valores de  $F_z$  obtidos para este problema pela invenção presente e a programação genética (Tab.1), mostra que a invenção presente é superior à programação genética em 374 vezes ( $5610000 / 15000$ ), portanto, mais de duas ordens de grandeza superior.

## EMPLILHAMENTO DE BLOCOS



Este problema de planeamento foi escolhido por ser um dos problemas de eleição na área da inteligência artificial e por ser considerado um problema sofisticado.

No problema do empilhamento de blocos existe uma série de blocos (as letras da palavra "universal", por exemplo) distribuídas aleatoriamente entre uma mesa e uma gaveta. As letras sobre a mesa estão espalhadas e qualquer letra pode ser recolhida, enquanto que na gaveta só se tem acesso à letra da frente e só é possível remover essa letra ou colocar outras letras à sua frente.

O objectivo deste problema consiste em descobrir um plano que, em presença de qualquer estado inicial de configurações de letras distribuídas aleatoriamente entre a mesa e a gaveta, consiga colocar as letras na gaveta pela mesma ordem em que aparecem na palavra "universal".

As funções e terminais escolhidos para resolver este problem, consistem num conjunto de acções e sensores. O conjunto de acções consiste nas funções {C, R, N, A} (*carregar da mesa para a gaveta, remover da gaveta para a mesa, NOT, e fazer até verdadeiro*, respectivamente), tomando as três primeiras 1 argumento e a última 2 argumentos. O conjunto de sensores consiste nos terminais {u, t, p} (*última letra da gaveta, letra do topo da gaveta correcta, próxima letra necessária*, respectivamente). A letra do topo correcta (t) refere-se somente à letra da frente da gaveta e se está correcta ou não; se a gaveta estiver vazia ou tiver algumas letras, todas elas colocadas correctamente, o sensor devolve Verdadeiro, de outro modo devolve Falso. A próxima letra necessária (p) refere-se obviamente à letra imediatamente a seguir a 't'.

Todos os problemas que envolvam uma acção iterativa como o 'A' (*fazer até verdadeiro*) levantam alguns problemas relacionados com os recursos de memória disponíveis num computador. Assim, têm que ser estabelecidas algumas regras de funcionamento para estas funções. Na invenção presente, cada 'A' é executado uma única vez, repetindo-se a acção num máximo de 20

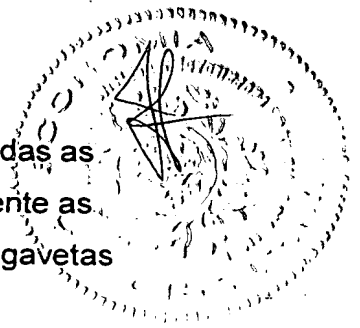
vezes, findas as quais aborta; os 'A' de cada sub-árvore são as primeiras acções executadas, sendo executadas numa ordem determinada: de baixo para cima e da esquerda para a direita.



Os casos de aptidão consistem, na invenção presente, em 10 casos iniciais de configurações, sendo 9 delas aleatórias e 1 em que as letras estão todas sobre a mesa ficando a gaveta obviamente vazia. O papel desta última configuração limita-se a impedir que algumas experiências abortem logo à partida, pois na verdade a invenção presente é capaz de descobrir um plano para resolver este problema usando 10 configurações totalmente aleatórias. Esta é de facto uma característica que diferencia a invenção presente da programação genética, uma vez que a programação genética só consegue resolver este problema usando 167 configurações inteligentemente arquitectadas (10 casos com zero a 9 letras correctamente colocadas na gaveta e as restantes sobre a mesa; 9 casos com zero a 7 letras correctamente colocadas na gaveta e exactamente uma letra incorrecta na frente com as restantes letras sobre a mesa; e 148 casos aleatórios)

Com os símbolos das funções e sensores foi criada uma população inicial de 30 cromossomas compostos de 3 genes de 9 de tamanho. Para este problema as sub-árvores são executadas sequencialmente, por exemplo, se a primeira sub-árvore esvazia as gavetas, a segunda pode proceder com o preenchimento parcial das gavetas, e a terceira pode acabar de preencher as gavetas. Para assinalar este facto, as árvores de expressão aparecem ligadas por um sinal '+' que representa exclusivamente a ordem sequencial na expressão e que, como todos os símbolos usados para ligar as sub-árvores, é extracromossómico. Os cromossomas foram posteriormente expressos e as suas aptidões determinadas no ambiente de selecção das 10 configurações iniciais. A aptidão foi calculada da seguinte forma: por cada gaveta vazia atribuiu-se um ponto de aptidão; por cada gaveta parcial e correctamente arrumada atribuíram-se 2 pontos de aptidão; e por cada gaveta correcta e completamente arrumada, atribuíram-se 3 pontos de aptidão. Assim, a aptidão máxima corresponde a 30 pontos. A ideia subjacente a este desenho, consiste em mostrar que a invenção presente descobre soluções movendo-se hierarquicamente em direcção a um plano perfeito. De facto, a

invenção presente descobre primeiro um plano capaz de esvaziar todas as gavetas, depois descobre a forma de encher correcta mas parcialmente as gavetas previamente esvaziadas e, por último, aprende a encher as gavetas correcta e totalmente.



A Fig. 14 mostra uma população inicial 1403 de cromossomas aleatórios criada numa experiência. Repare-se que dos indivíduos criados na população inicial, 17 deles têm aptidões positivas. Neste caso, não surgiu nenhum plano útil entre os indivíduos da população inicial, tendo os indivíduos 1 ou 2 pontos de aptidão, o que significa que não fazem nada e têm um ponto devido à gaveta vazia 9 (1402), ou então conseguem remover uma letra da gaveta, ficando neste caso com 2 pontos: 1 pela gaveta vazia 9 (1402) e outro ponto pela gaveta 4 (1401) que só tinha uma letra incorrecta e que ficou vazia quando esta foi removida.

Na geração seguinte foi descoberto o primeiro plano útil (1503) para resolver o problema (Fig. 15). Este plano remove todas as letras erradas da gaveta, conseguindo 11 pontos de aptidão para o caso concreto de configurações iniciais 1501: 2 pontos pela gaveta 6 1502 que fica com uma letra correcta e 9 pontos pelas restantes gavetas que ficam todas vazias. Repare-se que a primeira sub-árvore 1504 e a última sub-árvore 1506 não contribuem com nada, sendo toda a acção desempenhada pela segunda sub-árvore 1505.

Na geração 4 foi descoberto um plano mais avançado (1601) para resolver o problema (Fig. 16). Este plano não só remove todas as letras erradas da gaveta, como consegue colocar a primeira letra correcta em todas as gavetas, conseguindo 20 pontos de aptidão: 2 pontos por cada gaveta parcial e correctamente arrumada. Repare-se que a primeira 1602 e a segunda sub-árvore 1603 são homólogas, estando os genes respectivos duplicados. Estas sub-árvores são ambas capazes de remover todas as letras incorrectamente arrumadas. A última sub-árvore 1604 coloca uma letra em todas as gavetas vazias ou com letras correctas.

Na geração 13 foi descoberto um plano perfeito (1701) para resolver o problema (Fig. 17). Este plano remove primeiro todas as letras erradas da gaveta,

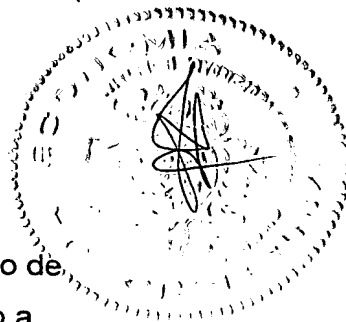
preenchendo depois todas as gavetas correctamente. Este plano é um plano universal para este problema e a aptidão tem o valor máximo de 30: 3 pontos por cada gaveta total e correctamente arrumada. Repare-se que a primeira sub-árvore 1702 não faz nada, sendo o plano executado pela segunda sub-árvore 1703 que remove todas as letras incorrectamente colocadas na gaveta, e pela terceira sub-árvore 1704 que acaba de preencher todas as gavetas com as letras correctas.

O cálculo da taxa de sucesso na resolução deste problema foi determinado avaliando 100 experiências idênticas à acima descrita, tendo sido obtido 0,7. Se compararmos o rendimento da invenção presente com o rendimento da programação genética na resolução deste problema, verifica-se que a invenção presente é 142 vezes superior à programação genética, apesar de a invenção presente utilizar 9 casos de configurações totalmente aleatórias num total de 10 configurações utilizadas como ambiente de selecção. Este facto merece destaque, pois em situações reais nem sempre é possível prever o tipo de casos que levarão o sistema a descobrir uma solução.

Na Tab. 2 é feita a comparação entre a invenção presente e a programação genética para uma corrida de 100 experiências idênticas no caso da invenção presente e uma corrida de 30 experiências no caso da programação genética (ver o cálculo do rendimento no exemplo da regressão simbólica).

Assim, a comparação dos valores de Fz obtidos para o problema do empilhamento de blocos pela invenção presente e a programação genética (Tab.2), mostra que a invenção presente é superior à programação genética em 142 vezes ( $17034000 / 120000$ ), logo mais de duas ordens de grandeza superior.

## MULTIPLEXER DE 6 BITS



O multiplexer de 6 bits é um circuito lógico muito utilizado no desenho de microprocessadores e em sistemas de telecomunicações, permitindo a serialização de canais paralelos de comunicação.

O objectivo do multiplexer de 6 bits consiste em descodificar um endereço de 2 bits (00, 01, 10, 11) e devolver o valor do registo correspondente ( $d_0, d_1, d_2, d_3$ ). Assim, o multiplexer de 6 bits é uma função de 6 argumentos: dois,  $a_0$  e  $a_1$ , determinam o endereço, e quatro,  $d_0$  a  $d_3$ , determinam o registo. Como a invenção presente utiliza cromossomas de caracteres, o conjunto de terminais consiste em  $\{a, b, 1, 2, 3, 4\}$ , que correspondem respectivamente a  $\{a_0, a_1, d_0, d_1, d_2, d_3\}$ .

Existem  $2^6=64$  combinações possíveis para os 6 argumentos do multiplexer de 6 bits e, neste caso, foram utilizadas as 64 combinações como casos de aptidão. Para determinar a aptidão, as 64 combinações foram agrupadas em 4 sub-grupos, cada um contendo as 16 combinações correspondentes a cada endereço. A aptidão de um programa é, neste caso, o número de casos de aptidão em que o valor devolvido é o correcto, mais um bónus de 84 pontos de aptidão por cada sub-grupo de combinações resolvido correctamente como um todo. Assim, por cada endereço descodificado é atribuído um total de 100 pontos ( $16 + 84$ ), sendo a aptidão máxima de 400 pontos. O que se pretende com este esquema, é que a invenção presente caminhe para a solução descobrindo um endereço de cada vez. E, de facto, a invenção presente aprende primeiro a descodificar um endereço, depois outro, e assim sucessivamente até ao último.

O conjunto de funções escolhidas para resolver este problema consistiu nas funções booleanas  $\{A, O, N\}$  (AND, OR e NOT, respectivamente, tomando as duas primeiras 2 argumentos e a última 1 argumento).

Com os símbolos das funções e dos terminais escolhidos foi criada uma população inicial de 250 cromossomas multigénicos compostos de 4 genes de 11 de tamanho. Para este problema as sub-árvores são ligadas pela função

comutativa OR. Os cromossomas foram posteriormente expressos e as suas aptidões determinadas para o conjunto de 64 casos de aptidão.



A Fig. 18 mostra o cromossoma do melhor indivíduo 1801 criado na população inicial, e a árvore de expressão respectiva 1802. Repare-se que este indivíduo consegue descodificar 44 dos 64 casos de aptidão, não resolvendo, no entanto, nenhum endereço na sua totalidade, não recebendo portanto nenhum bónus de aptidão.

Os indivíduos criados na população inicial são posteriormente seleccionados de acordo com a aptidão, acabando por se reproduzirem criando os indivíduos novos da geração seguinte. Este processo é repetido num total de 100 gerações.

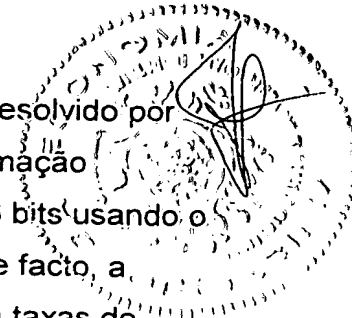
Na geração 4 foi criado um indivíduo capaz de decifrar completamente um endereço (16 casos) mais 32 casos distribuídos pelos restantes endereços, sendo a sua aptidão igual a 132. O cromossoma 1901 deste indivíduo e a respectiva árvore de expressão 1902 estão representados na Fig. 19.

Na geração 12 foi criado um indivíduo capaz de decifrar completamente dois endereços (32 casos) mais 16 casos distribuídos pelos restantes endereços, sendo a sua aptidão igual a 216. O cromossoma 2001 deste indivíduo e a respectiva árvore de expressão 2002 estão representados na Fig. 20.

Na geração 27 surge um indivíduo capaz de decifrar completamente três endereços (48 casos) mais 10 dos 16 casos do endereço por descodificar, sendo a sua aptidão igual a 310. O cromossoma 2101 deste indivíduo e a respectiva árvore de expressão 2102 estão representados na Fig. 21.

Na geração 86 surge um indivíduo capaz de decifrar completamente os quatro endereços do multiplexer de 6 bits (a totalidade dos 64 casos), tendo a aptidão máxima de 400. O cromossoma 2201 deste indivíduo e a respectiva árvore de expressão 2202 estão representados na Fig. 22 e, de facto, este programa é uma solução universal para o problema do multiplexer de 6 bits.

É importante salientar que o problema do multiplexer de 6 bits foi resolvido por outros sistemas de computação evolutiva, entre os quais a programação genética, mas nenhum deles conseguiu resolver o multiplexer de 6 bits usando o conjunto de funções escolhido neste exemplo (AND, OR, NOT). De facto, a invenção presente resolve o problema do multiplexer de 6 bits com taxas de sucesso de 100% usando a função booleana  $if(x,y,z)$  e o multiplexer de 11 bits com taxas de 57% usando a mesma função.



*Maria Cândida de Carvalho Ferreira*

(Doutora Maria Cândida de Carvalho Ferreira)

Angra do Heroísmo, 10 de Agosto de 2000



## REIVINDICAÇÕES

1ª



Um algoritmo genético para resolver problemas tais como optimização, descoberta de funções, planeamento e síntese lógica, usando uma população de indivíduos, caracterizado pelo facto de o cromossoma linear (entidade linear) dos indivíduos referidos ser de tamanho determinado e constituído por um ou mais genes simbólicos compostos por uma cabeça composta por símbolos que representam funções e argumentos e uma cauda composta por símbolos que representam argumentos, sendo o cromossoma referido expresso sob a forma de uma ou mais sub-entidades não-lineares de diferentes tamanhos e formas designadas sub-árvores de expressão, sendo as sub-árvores de expressão referidas ligadas por uma função escolhida formando uma árvore de expressão que é um arranjo hierárquico dos símbolos referidos representantes das funções e argumentos do algoritmo genético referido que engloba iterações duma série de passos, cada iteração englobando os seguintes passos:

expressão de cada cromossoma referido numa árvore de expressão referida;

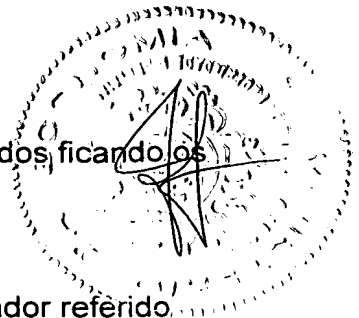
activação de cada árvore de expressão referida para produzir um resultado executando cada função de acordo com o arranjo hierárquico de funções e argumentos referido;

atribuição de um valor a cada resultado referido e associação de cada valor referido com a árvore de expressão referida correspondente que produziu cada resultado referido, sendo o valor referido indicativo da aptidão do indivíduo referido correspondente para resolver total ou parcialmente o problema;

selecção de pelo menos um indivíduo da população referida usando critérios de selecção, sendo os critérios de selecção referidos baseados nos valores associados referidos com cada indivíduo referido, preferindo os critérios de selecção referidos os indivíduos referidos com os valores associados referidos mais elevados;

replicação de tantos indivíduos seleccionados referidos quantos os indivíduos da população referida, em que cada indivíduo seleccionado referido reproduz

descendentes em número proporcional aos valores associados referidos, ficando os descendentes referidos inalterados;



escolha e execução de um ou vários operadores, em que cada operador referido escolhido pertence a um conjunto de operadores que inclui a mutação, transposição, inserção, transposição génica, recombinação pontual e recombinação em dois pontos;

se o operador referido escolhido for a mutação, o descendente referido é mutado sendo pelo menos um símbolo referido do cromossoma referido substituído por outro sem desfazer a organização estrutural e funcional da cabeça referida e da cauda referida dos genes referidos;

se o operador referido escolhido for a transposição, o descendente referido é modificado por transposição intracromossómica de elementos de transposição escolhidos aleatoriamente de entre os símbolos referidos da cabeça referida para o início do gene referido respectivo sem desfazer a organização estrutural e funcional da cabeça referida e da cauda referida dos genes referidos;

se o operador referido escolhido for a inserção, o descendente referido é modificado por inserção intracromossómica de elementos de inserção escolhidos aleatoriamente de entre os símbolos referidos do cromossoma referido para a cabeça referida do gene referido escolhido aleatoriamente sem desfazer a organização estrutural e funcional da cabeça referida e da cauda referida dos genes referidos;

se o operador referido escolhido for a transposição génica, o descendente referido é modificado por transposição intracromossómica de um gene completo escolhido aleatoriamente para o início do cromossoma referido escolhido aleatoriamente;

se o operador referido escolhido for a recombinação pontual, são escolhidos aleatoriamente pelo menos dois descendentes referidos e modificados por recombinação dos cromossomas referidos, trocando os cromossomas dos descendentes escolhidos partes dos cromossomas referidos respectivos entre si;

se o operador referido escolhido for a recombinação em dois pontos, são escolhidos aleatoriamente pelo menos dois descendentes referidos modificados por recombinação dos cromossomas referidos, trocando os cromossomas dos descendentes escolhidos um gene completo escolhido aleatoriamente entre si;

adição dos descendentes referidos à população referida;

## 2ª

Algoritmo genético como referido na reivindicação 1, caracterizado pelo facto de o passo de selecção referido englobar ulteriormente critérios de selecção baseados numa probabilidade que é proporcional ao valor associado referido com cada indivíduo dependendo, no entanto, da sorte na selecção.

## 3ª

Algoritmo genético como referido na reivindicação 1, caracterizado por englobar ulteriormente um passo de selecção e replicação do indivíduo referido com o valor associado referido mais elevado;

## 4ª

Algoritmo genético como referido na reivindicação 1, caracterizado pelo facto de um indivíduo da população referida que afigure um valor preestabelecido de aptidão na resolução do problema ser designado a solução do problema.

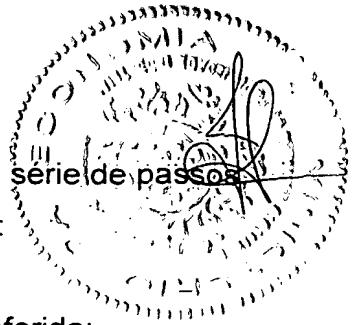
## 5ª

Algoritmo genético como referido na reivindicação 1, caracterizado pelo facto de a população inicial de indivíduos ser gerada aleatoriamente criando cromossomas com um ou mais genes compostos por cabeça e cauda em que a cabeça e a cauda referidas são um arranjo hierárquico de funções e argumentos utilizados para resolver o problema e a cauda referida é um arranjo hierárquico de argumentos referidos.

## 6ª

Num sistema de computação com uma população de programas expressos como árvores de expressão de vários tamanhos e formas, um algoritmo genético repetitivo para resolver problemas tais como optimização, descoberta de funções, planeamento e

síntese lógica, caracterizado pelo facto de englobar iterações numa série de passos;  
cada iteração do algoritmo genético referido englobando os passos:



expressão de cada programa referido numa árvore de expressão referida;

execução de cada programa referido para produzir um resultado;

atribuição de um valor a cada resultado referido e associação de cada valor referido com o programa referido correspondente que produziu cada resultado referido, sendo o valor referido indicativo da aptidão do programa referido correspondente para resolver total ou parcialmente o problema;

selecção de pelo menos um programa da população referida usando critérios de selecção, sendo os critérios de selecção referidos baseados nos valores associados referidos com cada programa referido, preferindo os critérios de selecção referidos os programas referidos com os valores associados referidos mais elevados;

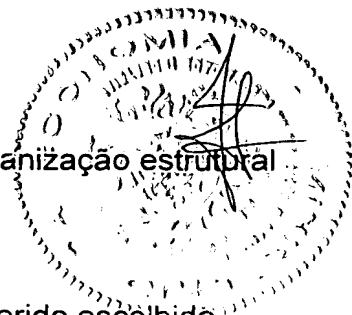
replicação de tantos programas seleccionados referidos quantos os programas da população referida, em que cada programa seleccionado referido reproduz programas novos em número proporcional aos valores associados referidos ficando os programas novos referidos inalterados;

escolha e execução de um ou vários operadores, em que cada operador referido escolhido pertence a um conjunto de operadores que inclui a mutação, transposição, inserção, transposição génica, recombinação pontual e recombinação em dois pontos;

se o operador referido escolhido for a mutação, o programa novo referido escolhido aleatoriamente é mutado sendo pelo menos um símbolo referido do programa referido substituído por outro sem desfazer a organização estrutural e funcional do gene referido do programa novo referido;

se o operador referido escolhido for a transposição, o programa novo referido escolhido aleatoriamente é modificado por transposição intracromossómica de elementos de transposição escolhidos aleatoriamente de entre os símbolos referidos da cabeça

referida para o início do gene referido respectivo sem desfazer a organização estrutural e funcional do gene referido do programa novo referido;



se o operador referido escolhido for a inserção, o programa novo referido escolhido aleatoriamente é modificado por inserção intracromossómica de elementos de inserção escolhidos aleatoriamente de entre os símbolos referidos do cromossoma referido para a cabeça referida do gene referido escolhido aleatoriamente sem desfazer a organização estrutural e funcional do gene referido do programa novo referido;

se o operador referido escolhido for a transposição génica, o programa novo referido escolhido aleatoriamente é modificado por transposição intracromossómica de um gene completo escolhido aleatoriamente para o início do cromossoma referido do programa novo referido;

se o operador referido escolhido for a recombinação pontual, são escolhidos aleatoriamente pelo menos dois programas novos referidos e modificados por recombinação dos cromossomas referidos dos programas novos referidos escolhidos, trocando os cromossomas dos programas novos referidos escolhidos partes dos cromossomas referidos respectivos entre si;

se o operador referido escolhido for a recombinação em dois pontos, são escolhidos aleatoriamente pelo menos dois programas novos referidos e modificados por recombinação dos cromossomas referidos dos programas novos referidos escolhidos, trocando os cromossomas dos programas novos referidos escolhidos um gene completo escolhido aleatoriamente entre si;

adição dos programas novos referidos à população referida;

### 7ª

Algoritmo genético como referido na reivindicação 6, caracterizado pelo facto de o passo de selecção referido englobar ulteriormente critérios de selecção baseados numa probabilidade que é proporcional ao valor associado referido com cada programa dependendo, no entanto, da sorte na selecção.

8ª

Algoritmo genético como referido na reivindicação 6, caracterizado por englobar ulteriormente um passo de selecção e replicação do programa referido com o valor associado referido mais elevado;



9ª

Algoritmo genético como referido na reivindicação 6, caracterizado pelo facto de um programa da população referida que afigure um valor preestabelecido de aptidão na resolução do problema ser designado a solução do problema.

10ª

Algoritmo genético como referido na reivindicação 6, caracterizado pelo facto de a população inicial de programas ser gerada aleatoriamente criando programas com cromossomas com um ou mais genes compostos por cabeça e cauda em que a cabeça e a cauda referidas são uma estrutura hierárquica de funções e argumentos utilizados para resolver o problema e a cauda referida é um arranjo hierárquico de argumentos referidos.

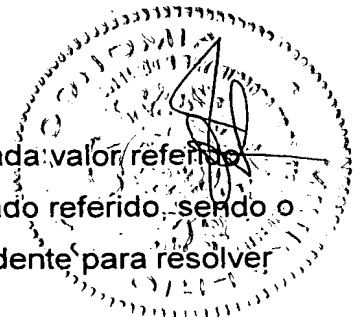
11ª

Num sistema de computação paralela com uma população de programas expressos como árvores de expressão de vários tamanhos e formas e em que mais do que um programa pode ser executado simultaneamente, um conjunto de algoritmos genéticos paralelos para resolver problemas tais como optimização, descoberta de funções, planeamento e síntese lógica, caracterizado pelo facto de mais do que um algoritmo genético paralelo do conjunto de algoritmos genéticos paralelos referido poder ser executado simultaneamente, cada algoritmo genético paralelo referido do conjunto de algoritmos genéticos paralelos referido contendo iterações de uma série de passos, cada iteração de cada algoritmo genético paralelo referido englobando os passos:

expressão de cada programa referido numa árvore de expressão referida;

execução de cada programa referido para produzir um resultado;

atribuição de um valor a cada resultado referido e associação de cada valor referido com o programa referido correspondente que produziu cada resultado referido, sendo o valor referido indicativo da aptidão do programa referido correspondente para resolver total ou parcialmente o problema;



selecção de pelo menos um programa da população referida usando critérios de selecção, sendo os critérios de selecção referidos baseados nos valores associados referidos com cada programa referido, preferindo os critérios de selecção referidos os programas referidos com os valores associados referidos mais elevados;

replicação de tantos programas seleccionados referidos quantos os programas da população referida, em que cada programa seleccionado referido reproduz programas novos em número proporcional aos valores associados referidos ficando os programas novos referidos inalterados;

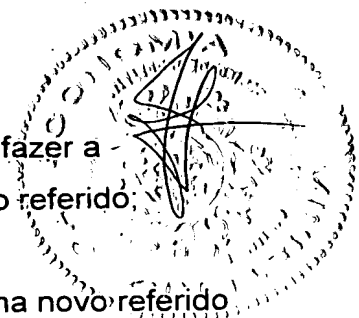
escolha e execução de um ou vários operadores, em que cada operador referido escolhido pertence a um conjunto de operadores que inclui a mutação, transposição, inserção, transposição génica, recombinação pontual e recombinação em dois pontos;

se o operador referido escolhido for a mutação, o programa novo referido escolhido aleatoriamente é mutado sendo pelo menos um símbolo referido do programa referido substituído por outro sem desfazer a organização estrutural e funcional do gene referido do programa novo referido;

se o operador referido escolhido for a transposição, o programa novo referido escolhido aleatoriamente é modificado por transposição intracromossómica de elementos de transposição escolhidos aleatoriamente de entre os símbolos referidos da cabeça referida para o início do gene referido respectivo sem desfazer a organização estrutural e funcional do gene referido do programa novo referido;

se o operador referido escolhido for a inserção, o programa novo referido escolhido aleatoriamente é modificado por inserção intracromossómica de elementos de inserção escolhidos aleatoriamente de entre os símbolos referidos do cromossoma referido para

a cabeça referida do gene referido escolhido aleatoriamente sem desfazer a organização estrutural e funcional do gene referido do programa novo referido;



se o operador referido escolhido for a transposição génica, o programa novo referido escolhido aleatoriamente é modificado por transposição intracromossómica de um gene completo escolhido aleatoriamente para o início do cromossoma referido do programa novo referido;

se o operador referido escolhido for a recombinação pontual, são escolhidos aleatoriamente pelo menos dois programas novos referidos e modificados por recombinação dos cromossomas referidos dos programas novos referidos escolhidos, trocando os cromossomas dos programas novos referidos escolhidos partes dos cromossomas referidos respectivos entre si;

se o operador referido escolhido for a recombinação em dois pontos, são escolhidos aleatoriamente pelo menos dois programas novos referidos e modificados por recombinação dos cromossomas referidos dos programas novos referidos escolhidos, trocando os cromossomas dos programas novos referidos escolhidos um gene completo escolhido aleatoriamente entre si;

adição dos programas novos referidos à população referida;

## 12ª

Algoritmo genético como referido na reivindicação 11, caracterizado pelo facto de o passo de selecção referido englobar ulteriormente critérios de selecção baseados numa probabilidade que é proporcional ao valor associado referido com cada programa dependendo, no entanto, da sorte na selecção.

## 13ª

Algoritmo genético como referido na reivindicação 11, caracterizado por englobar ulteriormente um passo de selecção e replicação do programa referido com o valor associado referido mais elevado;

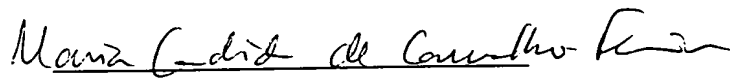


14<sup>a</sup>

Algoritmo genético como referido na reivindicação 11, caracterizado pelo facto de um programa da população referida que afixa um valor preestabelecido de aptidão na resolução do problema ser designado a solução do problema.

15<sup>a</sup>

Algoritmo genético como referido na reivindicação 11, caracterizado pelo facto de a população inicial de programas ser gerada aleatoriamente criando programas com cromossomas com um ou mais genes compostos por cabeça e cauda em que a cabeça e a cauda referidas são uma estrutura hierárquica de funções e argumentos utilizados para resolver o problema e a cauda referida é um arranjo hierárquico de argumentos referidos.



(Doutora Maria Cândida de Carvalho Ferreira)

Angra do Heroísmo, 10 de Agosto de 2000

**Figura 1**

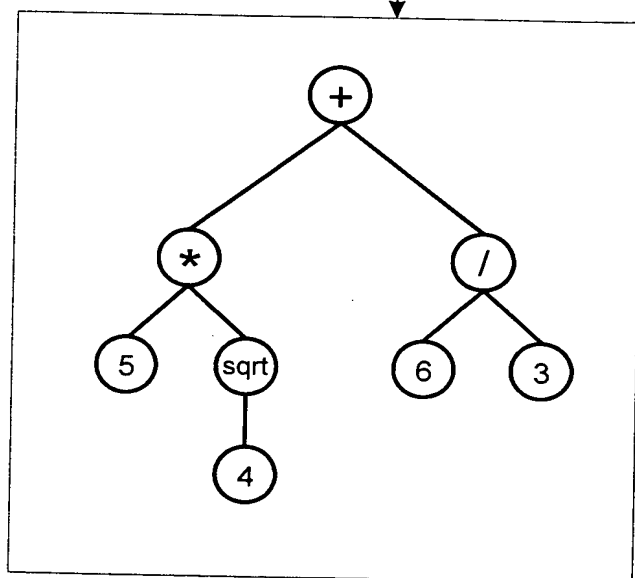
S-expressão em LISP

102

$(+ (* 5 (\text{sqrt } 4)) (/ 63))$

Representação em árvore

103



Matemática convencional

101

$5 \cdot \sqrt{4} + \frac{6}{3}$

Cromossoma da invenção presente

104

+ \* / 5 0 6 3 4

105

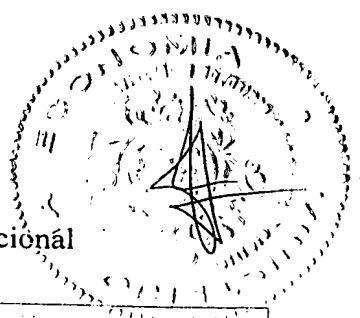
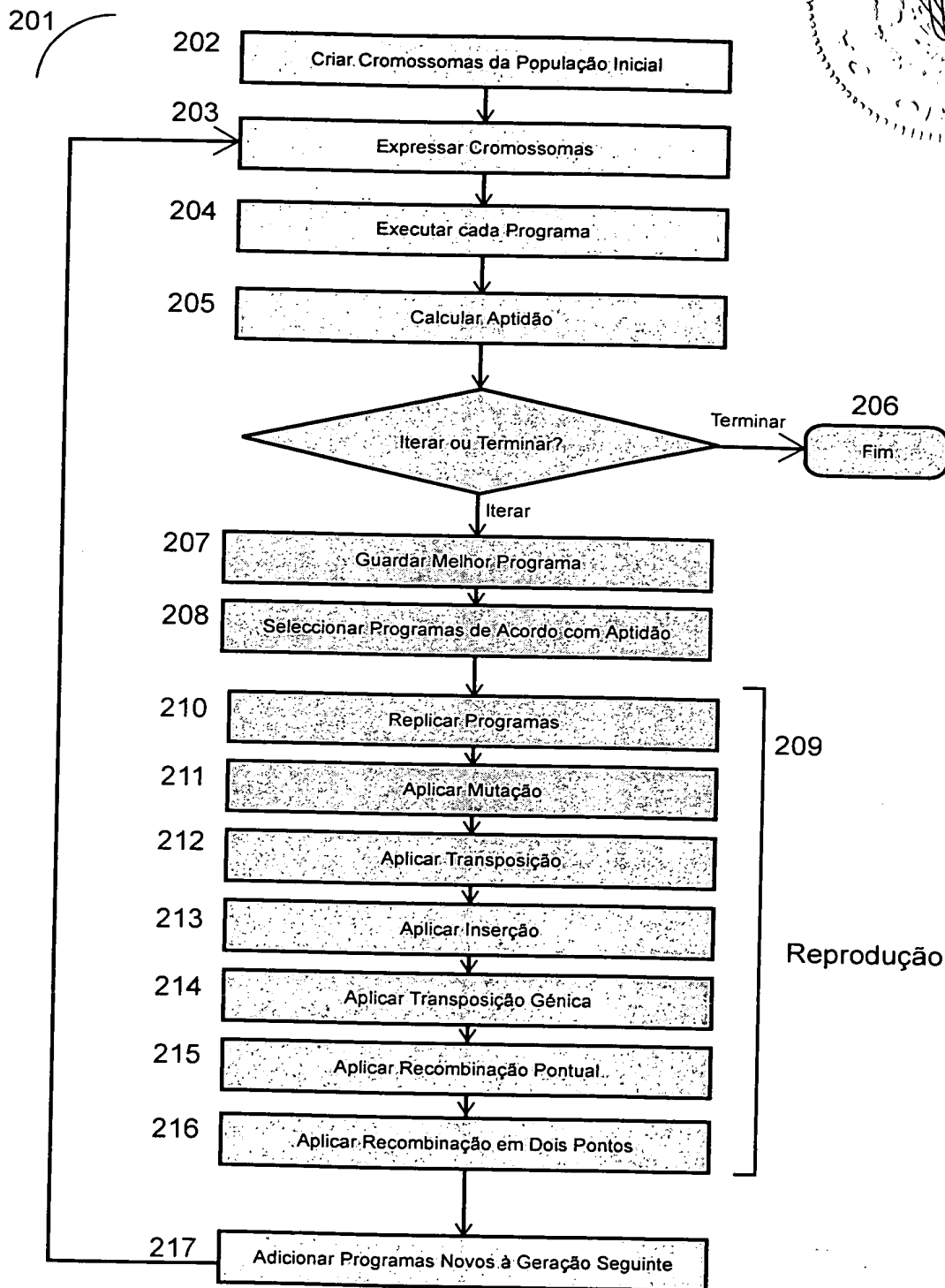


Figura 2



**Figura 3**

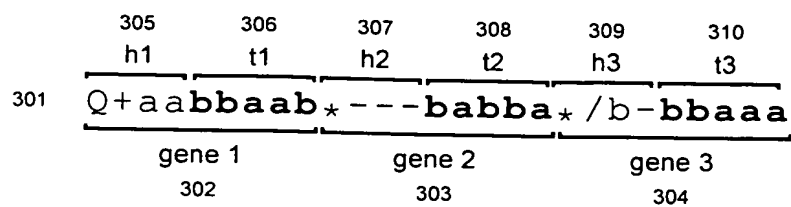


Figura 4

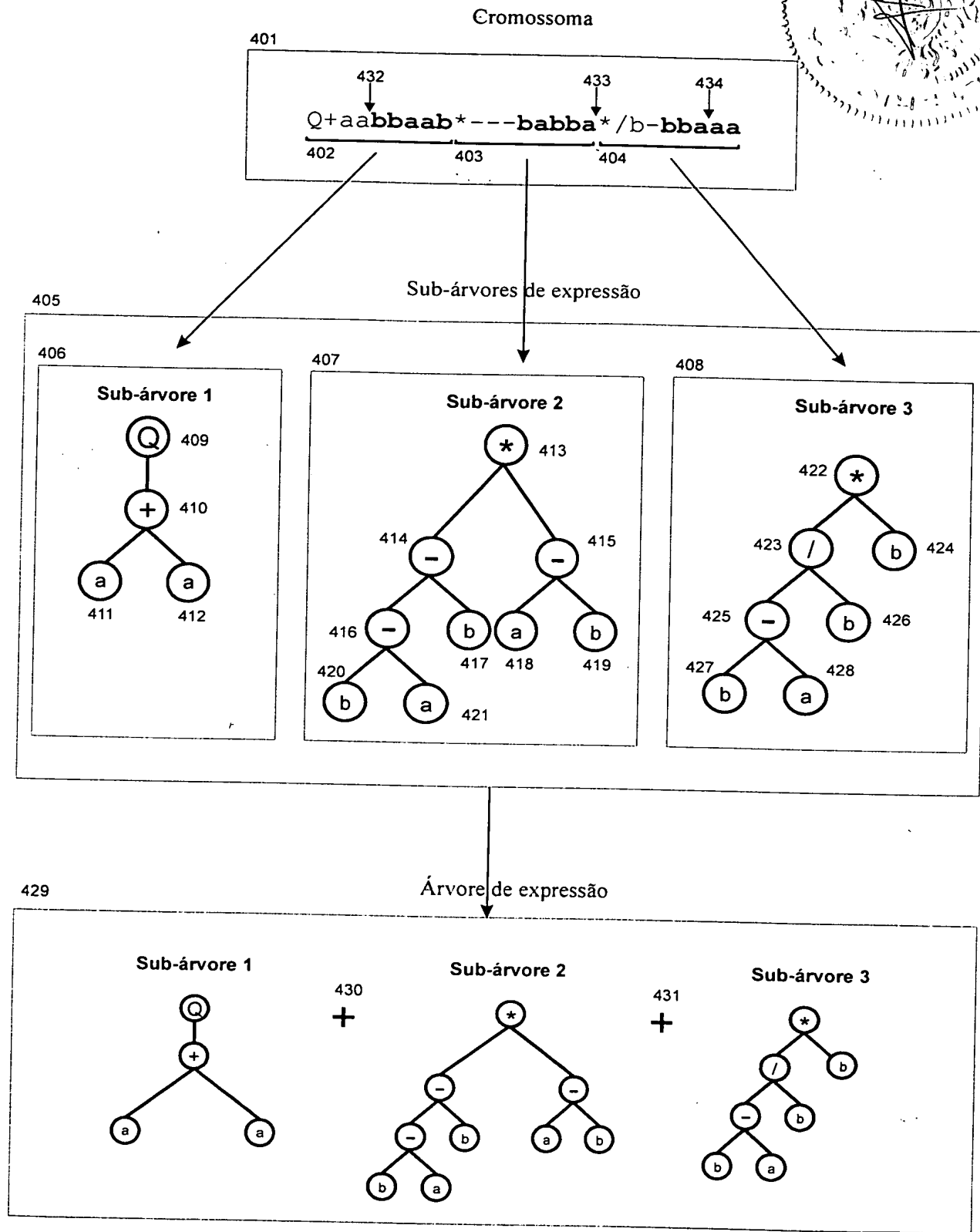
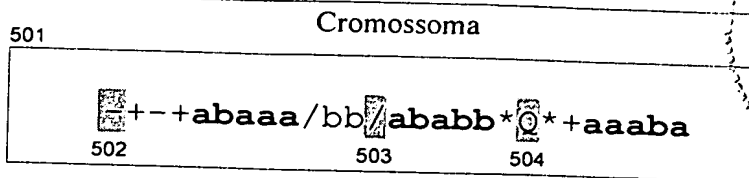
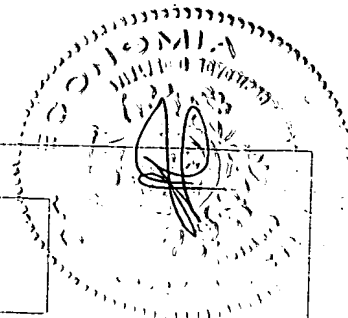
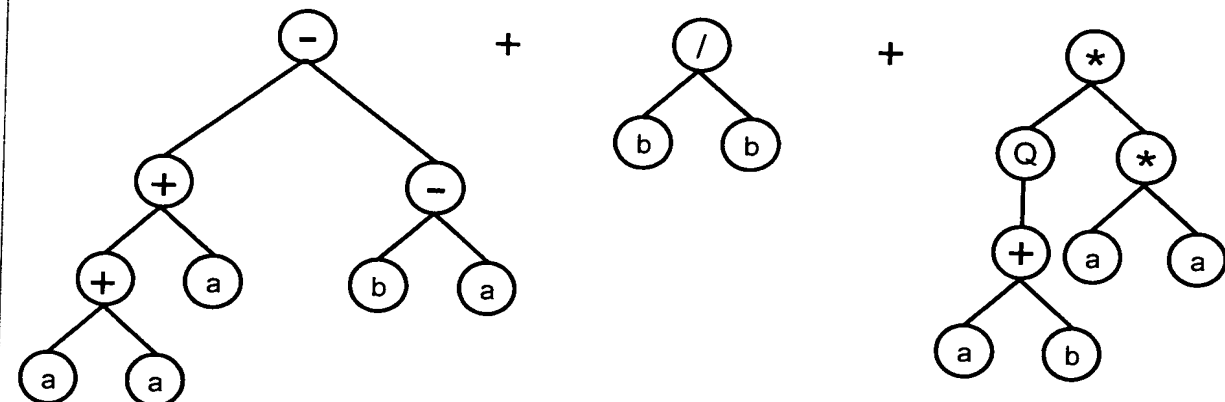


Figura 5

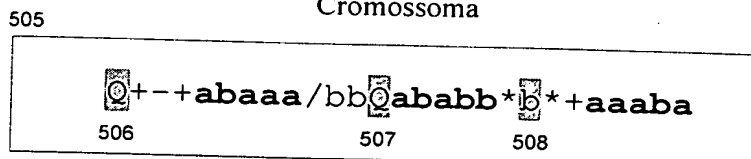


509

Árvore de expressão



Cromossoma



510

Árvore de expressão

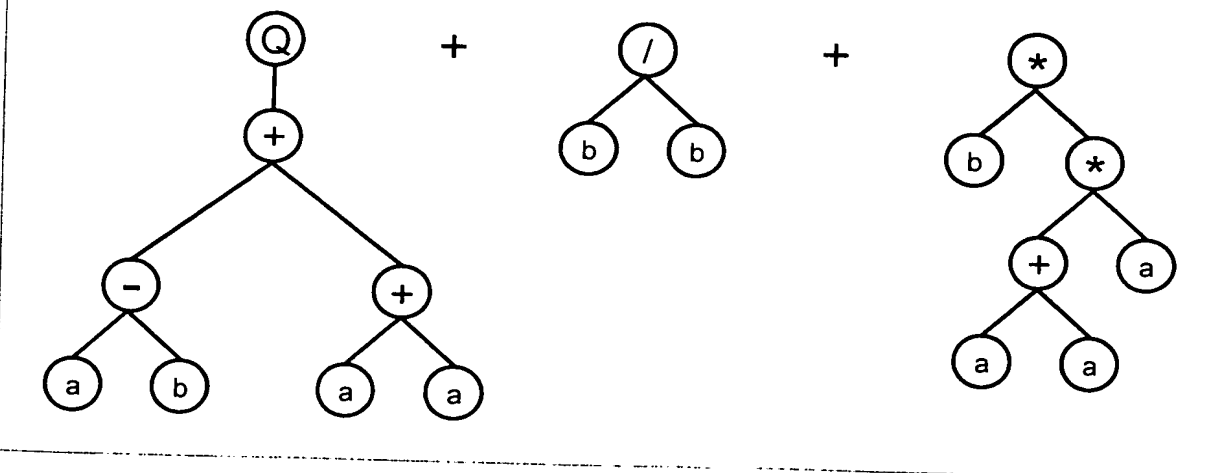


Figura 6

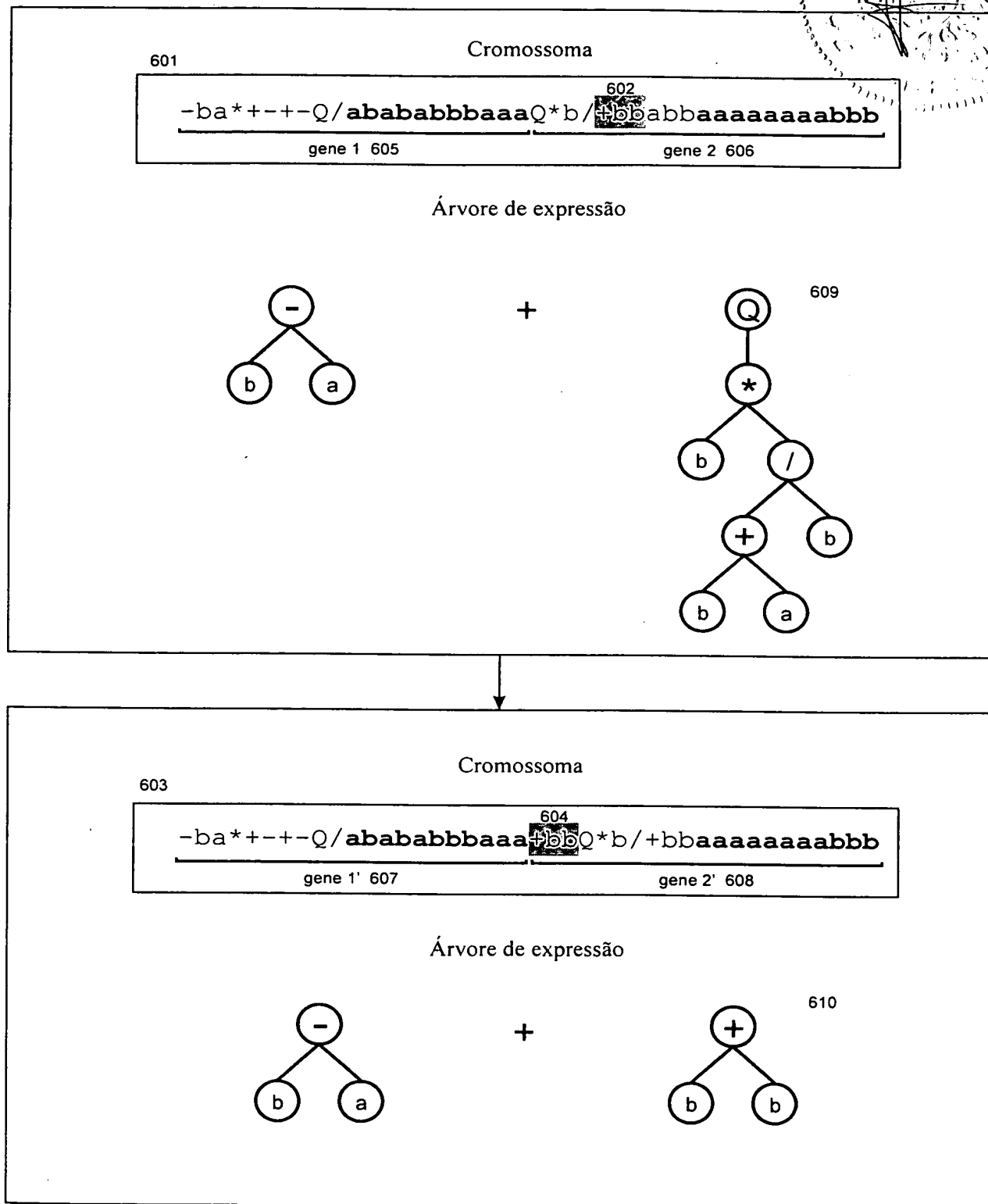


Figura 7

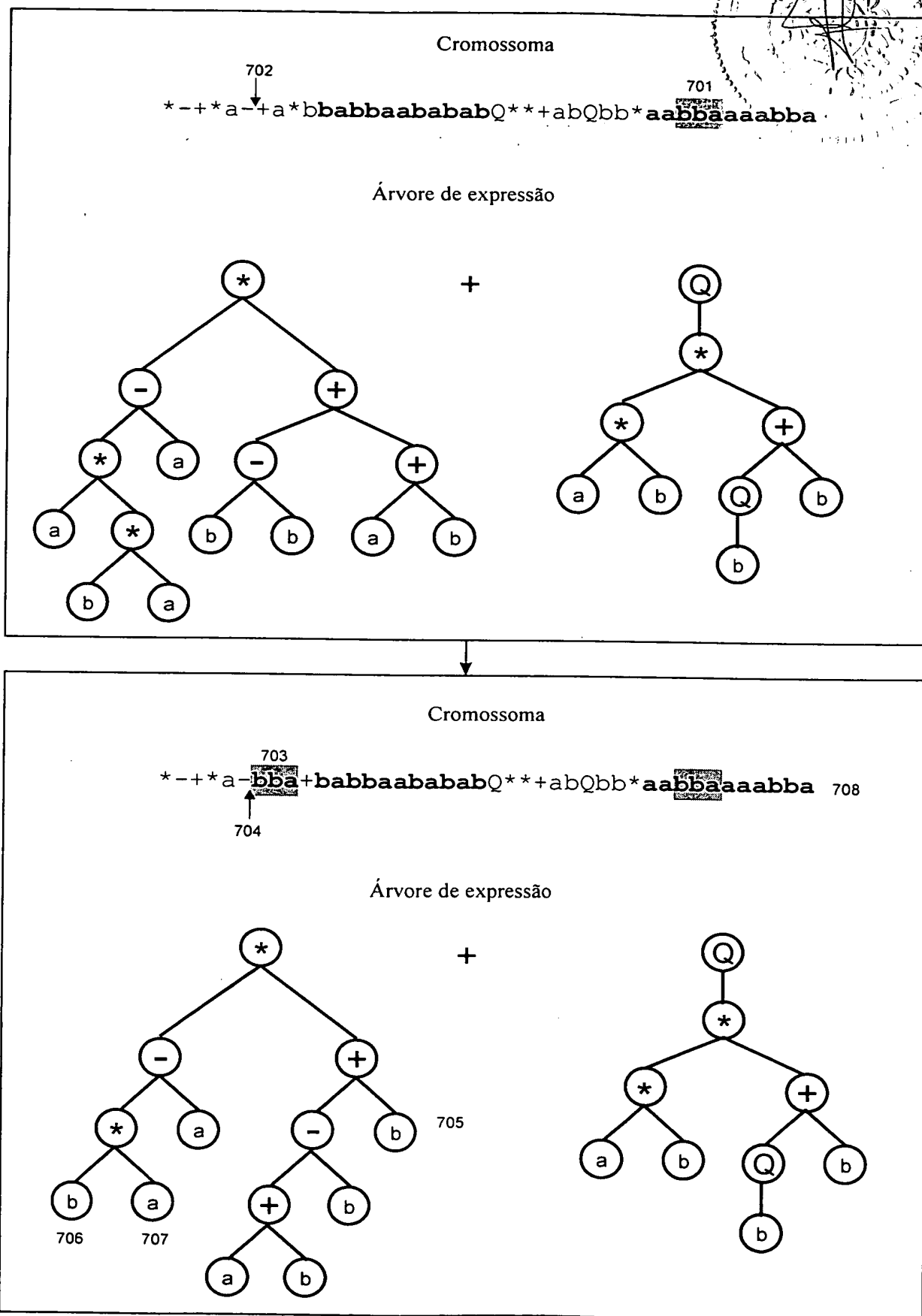




Figura 8

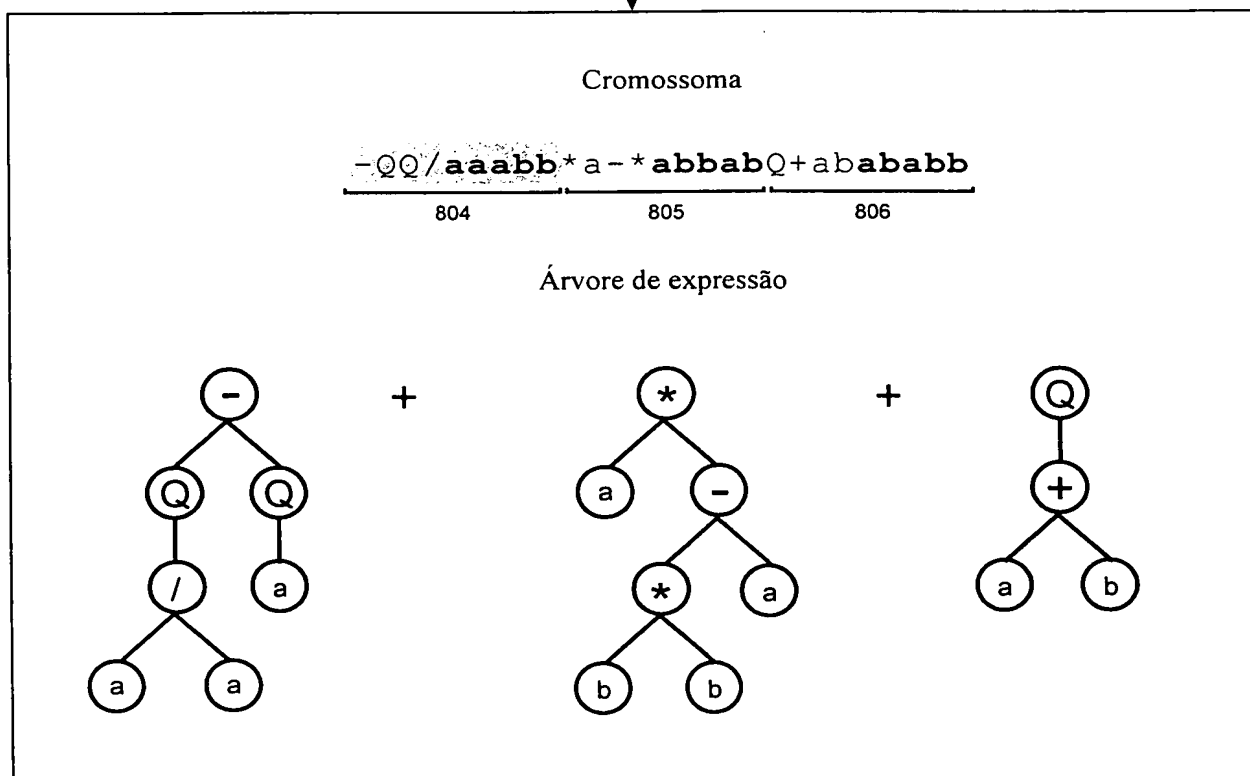
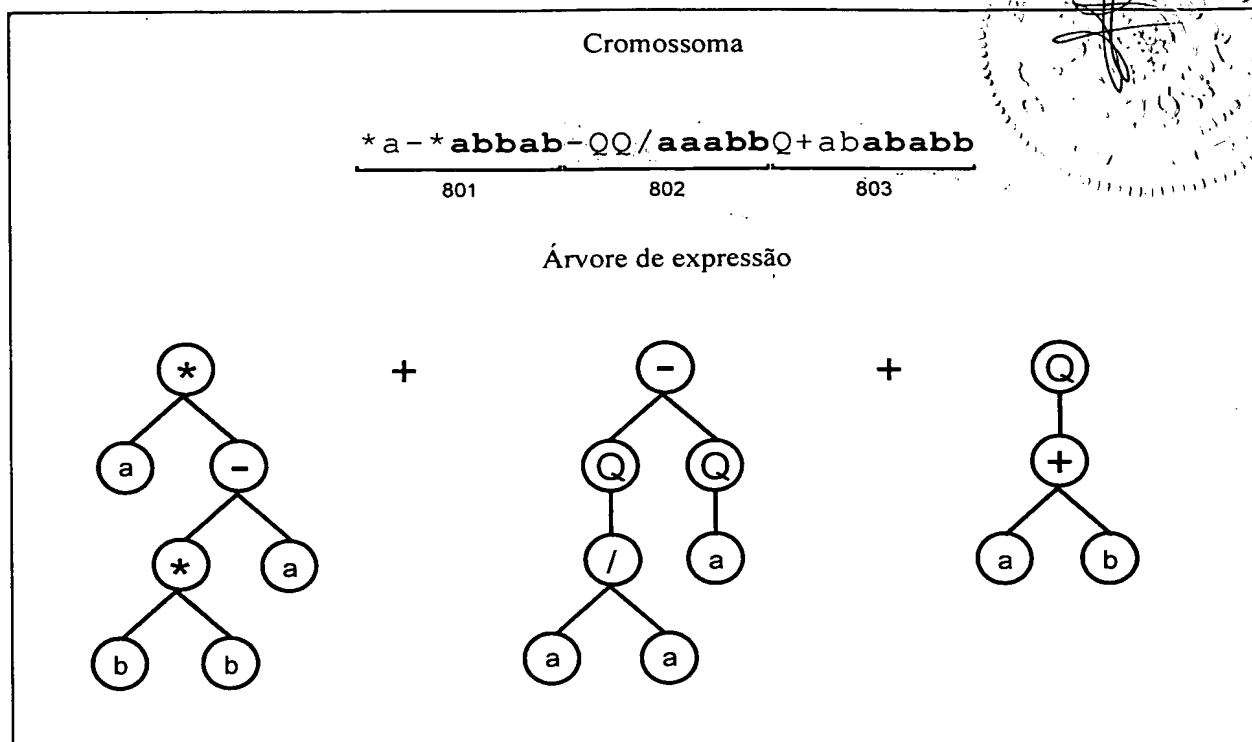


Figura 9

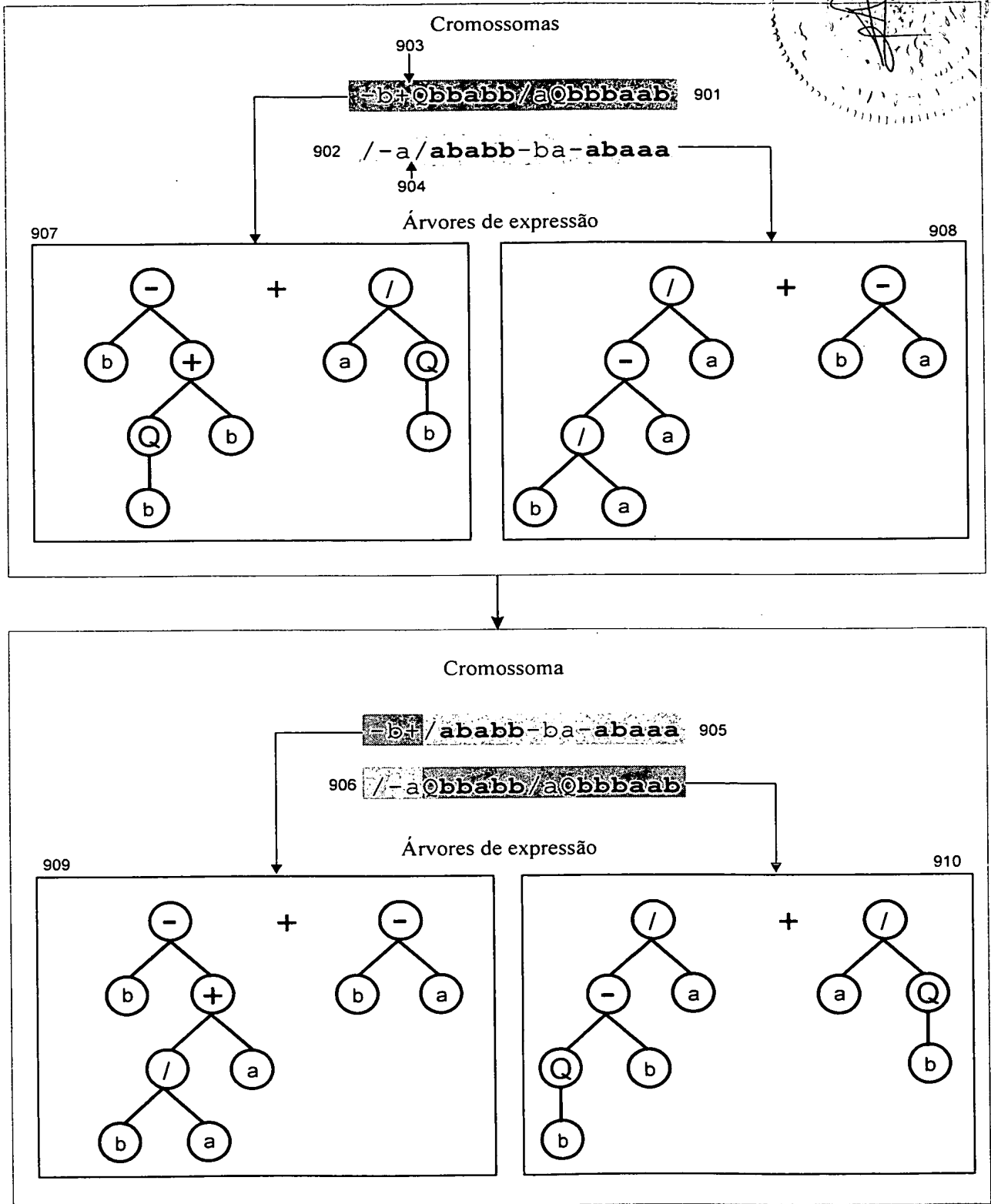


Figura 10a

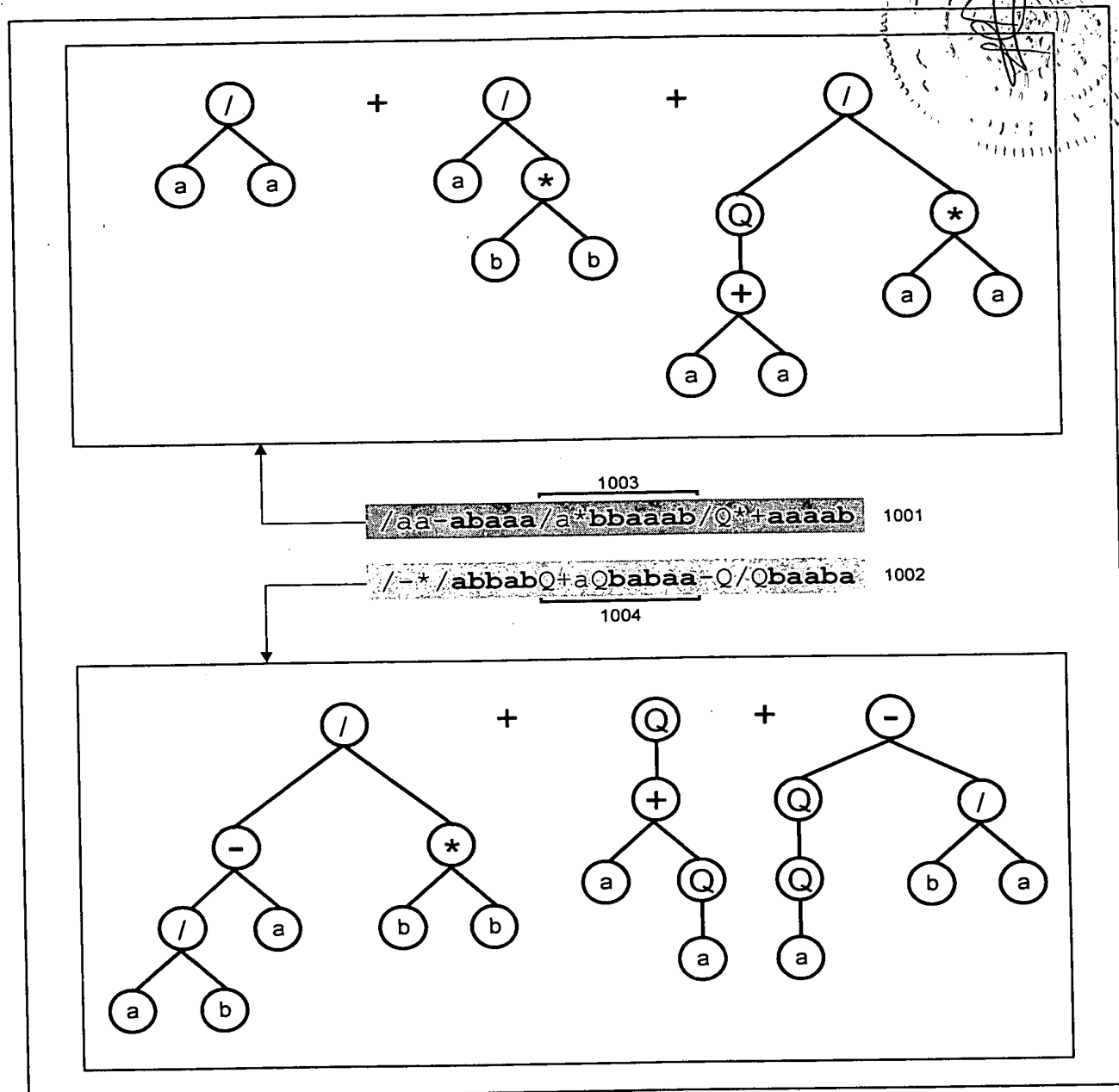


Figura 10a

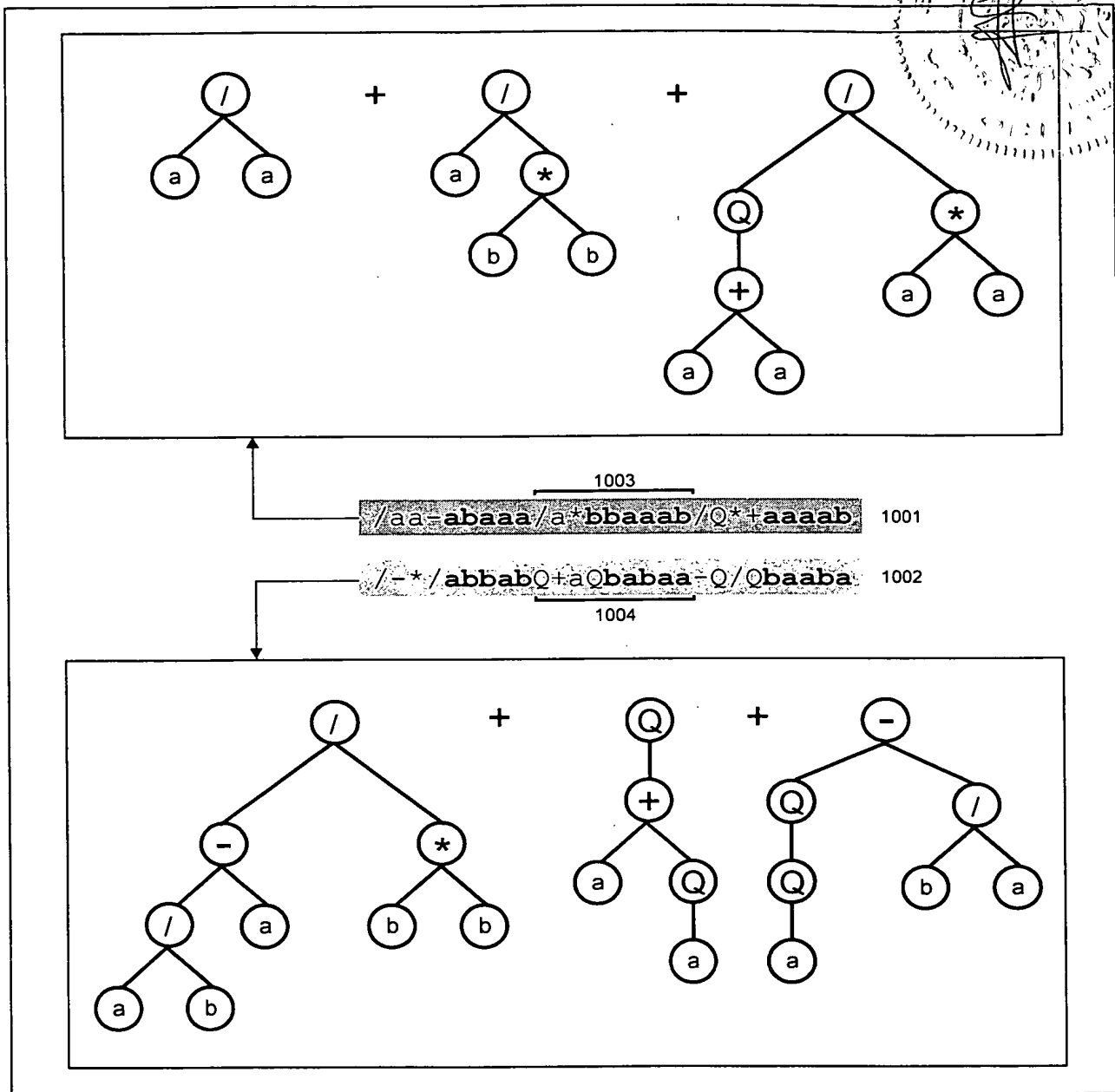
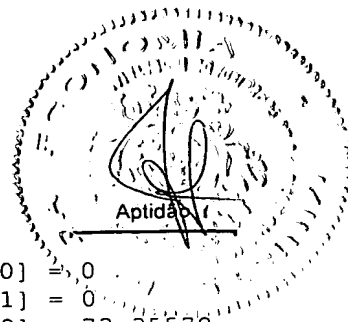


Figura 11

1101

Cromossomas



```
//*/-aaaaaa---+aaaaaa*++*+aaaaaa*++*+aaaaaa-[ 0] = 0
//+//aaaaaa+/*+/aaaaaa*-a*+aaaaaa-a**+aaaaaa-[ 1] = 0
+a*+aaaaaa+---+aaaaaa*---aaaaaa*++*+aaaaaa-[ 2] = 73.35578
+a*-aaaaaa/+a/aaaaaa-/*+aaaaaa**/-aaaaaa-[ 3] = 0
/+/*+aaaaaa/a/+aaaaaa+/aa+aaaaaa*/-/+aaaaaa-[ 4] = 26.6697
++/-aaaaaa/-*/aaaaaa*++*+aaaaaa/a*a-aaaaaa-[ 5] = 0
+*+//aaaaaa-a*+aaaaaa-/*//aaaaaa/+/*+aaaaaa-[ 6] = 25.44238
*--*-aaaaaa+a**+aaaaaa-a**+aaaaaa+/-*-aaaaaa-[ 7] = 0
-*a+-aaaaaa*a**+aaaaaa+*-a-aaaaaa//++*+aaaaaa-[ 8] = 22.67557
/+---+aaaaaa*a/*+aaaaaa/aa*+aaaaaa-*/a-aaaaaa-[ 9] = 0
*a/a+aaaaaa*++/-aaaaaa*+/*//aaaaaa-a*-aaaaaa-[10] = 35.0658
-a*-aaaaaa*++*+aaaaaa-+/-aaaaaa*a/+aaaaaa-[11] = 97.6903
/aa*+aaaaaa*+//+aaaaaa/*+*+aaaaaa*++*+aaaaaa-[12] = 45.73774
*-a-aaaaaa+/*+aaaaaa*--*+aaaaaa---a-aaaaaa-[13] = 0
++*/-aaaaaa-a/aaaaaa*-/++*+aaaaaa/+---+aaaaaa-[14] = 0
+*+//aaaaaa*/*//aaaaaa/a-aaaaaa---+*+aaaaaa-[15] = 0
/aa--aaaaaa-+*-/aaaaaa+*+a/aaaaaa-/a+aaaaaa-[16] = 7.7575
++*--aaaaaa+*+---aaaaaa+a-+*+aaaaaa/a*-aaaaaa-[17] = 0
*++*+aaaaaa-/+*-aaaaaa-a/a+aaaaaa*+*-aaaaaa-[18] = 0
+a-+*+aaaaaa/a/+aaaaaa//+*-aaaaaa/a*/aaaaaa-[19] = 0
--a+-aaaaaa*a/a/aaaaaa+*-/+aaaaaa*a+/-aaaaaa-[20] = 21.5497
/+**+aaaaaa++a/+aaaaaa*+---aaaaaa/+*+aaaaaa-[21] = 18.06512
/+aa-aaaaaa+//+*+aaaaaa*a+/-aaaaaa-*/-aaaaaa-[22] = 17.4636
/-/+---aaaaaa/-+*+aaaaaa/a/-*+aaaaaa-+*+aaaaaa-[23] = 0
++*-aaaaaa/*a*-aaaaaa/-+---aaaaaa+/-*/aaaaaa-[24] = 27.9458
-/----aaaaaa-a*+aaaaaa+---+aaaaaa-+a+aaaaaa-[25] = 0
+---+aaaaaa/+/+aaaaaa-a/*+aaaaaa+*-a/aaaaaa-[26] = 18.2736
-*+a+aaaaaa/-/+aaaaaa*+aa*+aaaaaa/-/a-aaaaaa-[27] = 80.0881
/-+++aaaaaa*+/-/aaaaaa/-+*+aaaaaa/-*+*+aaaaaa-[28] = 0
+*+//aaaaaa*----+aaaaaa/a/+aaaaaa*a+//+aaaaaa-[29] = 31.31912
```

Número do cromossoma

Figura 12

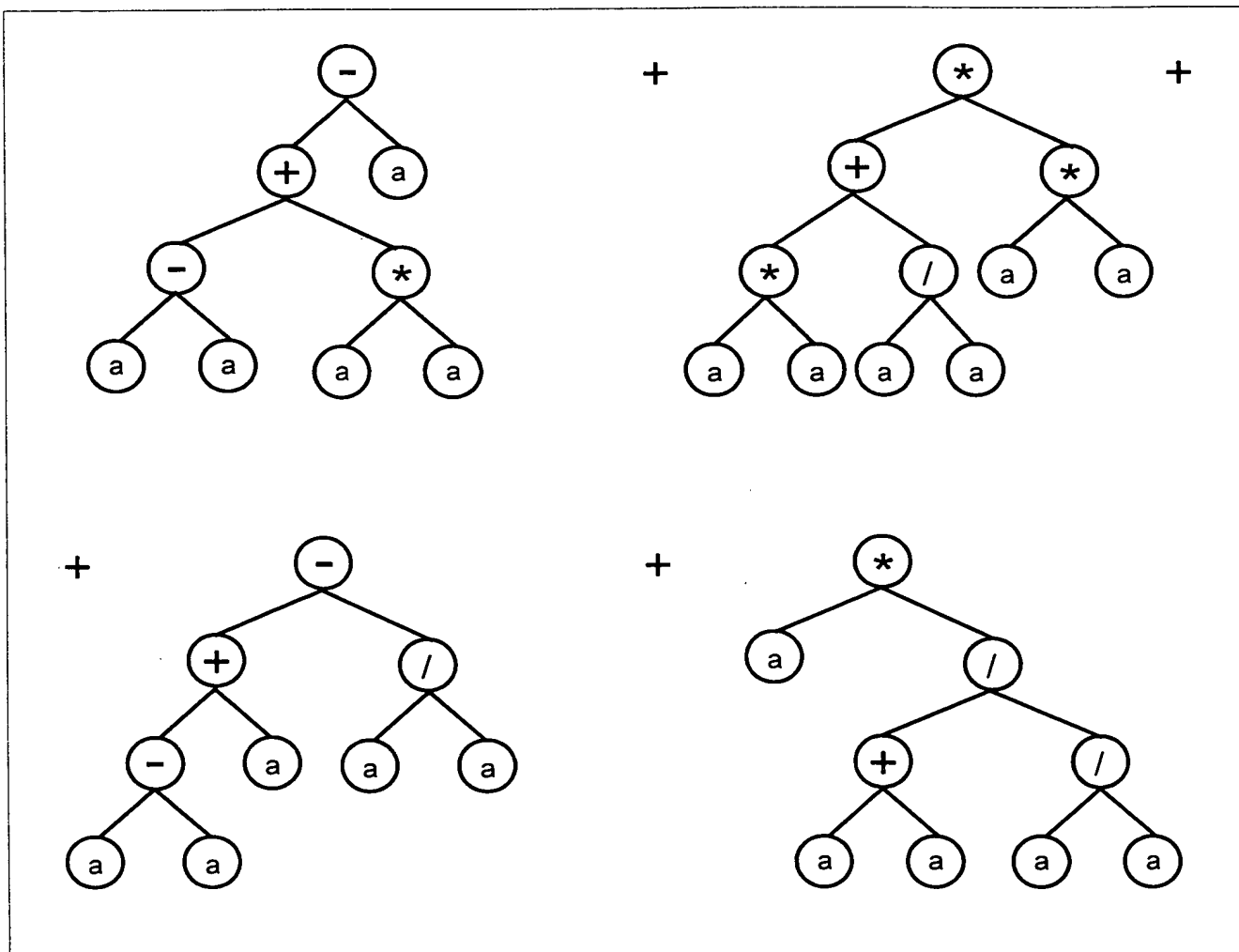
Cromossoma

1201

-+a-\*aaaaaa\*+\*\*/aaaaaa-+/-aaaaaaa\*a/+ /aaaaaa

Árvore de expressão

1202



Expressão matemática

1203

$$y = (a^2 - a) + (a^4 + a^2) + (a - 1) + (2a^2) = a^4 + 4a^2 - 1$$

Figura 13

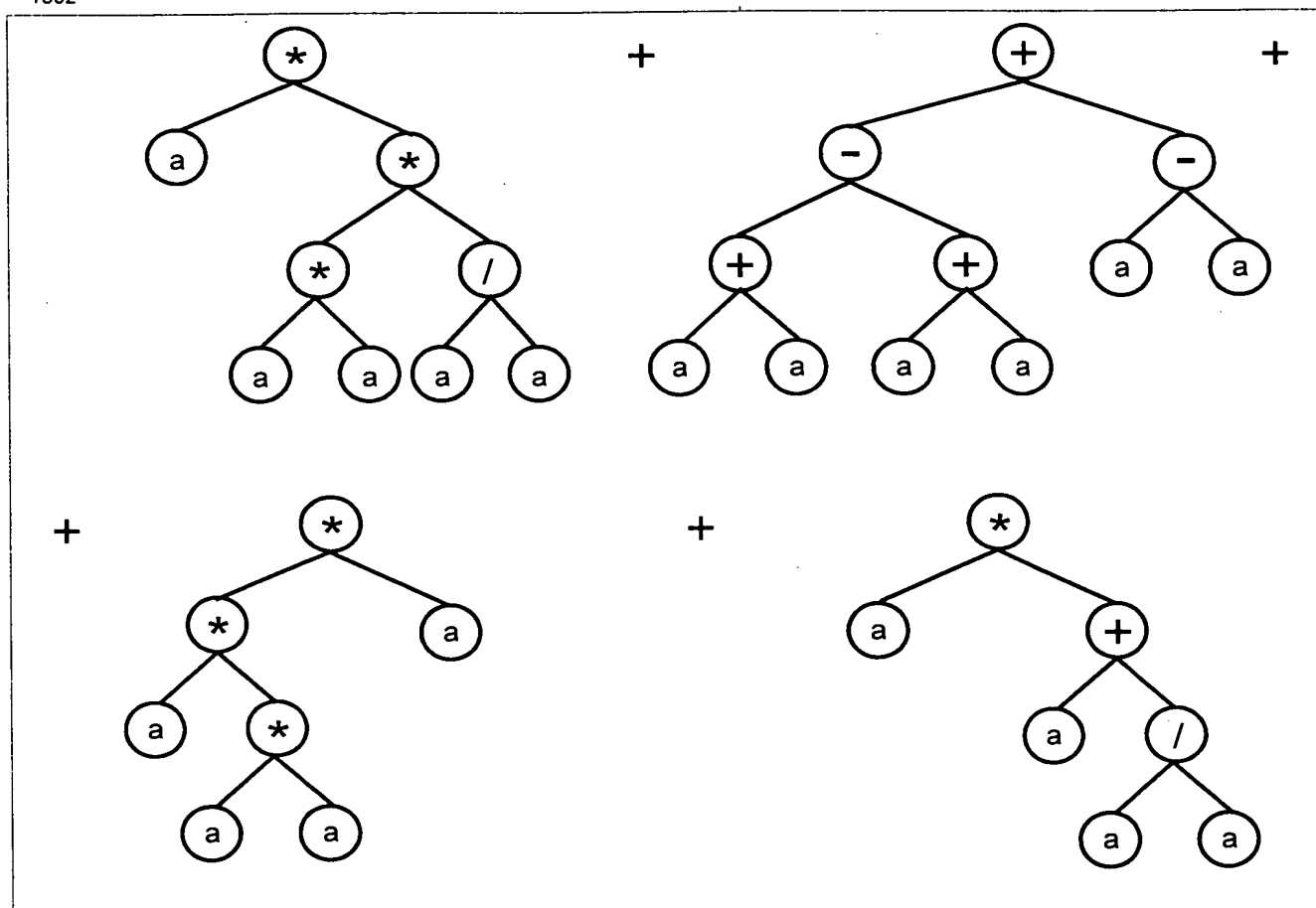
Cromossoma

1301

\*a\*\* / aaaaaa+--+aaaaaa\*\*aa\*aaaaaa\*a+a/aaaaaa...

Árvore de expressão

1302



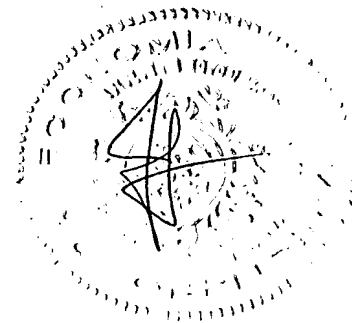
Expressão matemática

1303

$$y = (a^3) + (0) + (a^4) + (a^2 + a) = a^4 + a^3 + a^2 + a$$

**Tabela 1**

	Invenção Presente	Programação Genética
G	50	51
P	30	500
C	10	20
P <sub>s</sub>	1	0,35
R <sub>z</sub>	1	11
F <sub>z</sub>	15000	5610000





**Figura 14**

Gavetas do estado inicial:

```
luaaine###-0
unervsi##-1
auvse###-2
nalvuesri-3
s#####-4 1401
sluavn###-5
esiv#####-6
narsv#####-7
vlera#####-8
#####-9 1402
```

1403	Cromossomas	Aptidão
	ApAAtuputCuRuputptAttCtptuu-[ 0] = 1	
	ARRpttutpAAupupuupNCRNuttu-[ 1] = 0	
	CCRApttttCApNuptppRNAAttpup-[ 2] = 1	
	ANpNttuptCARptutuuNRpAtttut-[ 3] = 1	
	ARCRtpptuRuuAppptpAttptutp-[ 4] = 1	
	ApuApuuutAARCutupuACNptpppu-[ 5] = 0	
	RARuputppCACAppttupRRCNttutu-[ 6] = 0	
	AtAuppppuCuAtppptuNACAtttpp-[ 7] = 1	
	NtNAuputpACutppptuCuCRptppu-[ 8] = 0	
	NtANpptutAuRptpppuRpNAutupu-[ 9] = 1	
	AuutttttuRptRptttuRAARutupt-[10] = 0	
	CpNRtuuupCCCNpupptRptCuptuu-[11] = 2	
	RACAuptutCACtutuupNRtNptput-[12] = 1	
	AApNuuttpANCuptutuRCAutptup-[13] = 0	
	CtCututtuCAAAutptuANRNuttpt-[14] = 0	
	CAApptuttCAtNutppuAtpAtutut-[15] = 1	
	NpAAuppupAAAuptpuuAttRtputt-[16] = 1	
	AARttttuuuNAARtpuuAutuutptp-[17] = 1	
	ApRAtutptNNAAppppuACtRptuup-[18] = 0	
	AAApppputRNACupptpACNttuptu-[19] = 0	
	CpRNppppuACANttuttNAAAutput-[20] = 2	
	AtNAututtAuptttuupARRCtuppp-[21] = 0	
	CAAAtputuAtANptpupAAptpuut-[22] = 1	
	ARNRtuuupApAtttputApRNupuut-[23] = 0	
	RtNNtpuppCtAuupppuCAANpuutu-[24] = 0	
	RCAtuupttAutAptutpAAAtttuu-[25] = 0	
	RtAuputtAAApptutRpRpptpuu-[26] = 1	
	CpAAAtputuCCpNpttutAuuRppupt-[27] = 2	
	AACRtpupuRuAAttuptCAuuppuuu-[28] = 2	
	RuAAputtuRACNuupptAuRppuupp-[29] = 2	

Número do cromossoma

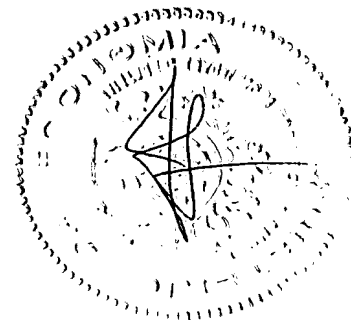


Figura 15

Gavetas do estado inicial:

1501

rilasnvu#-0
rls#####-1
anruievl-2
nr#####-3
viaslre##-4
ievlanru#-5
uenari###-6
neai#####-7
li#####-8
#####-9

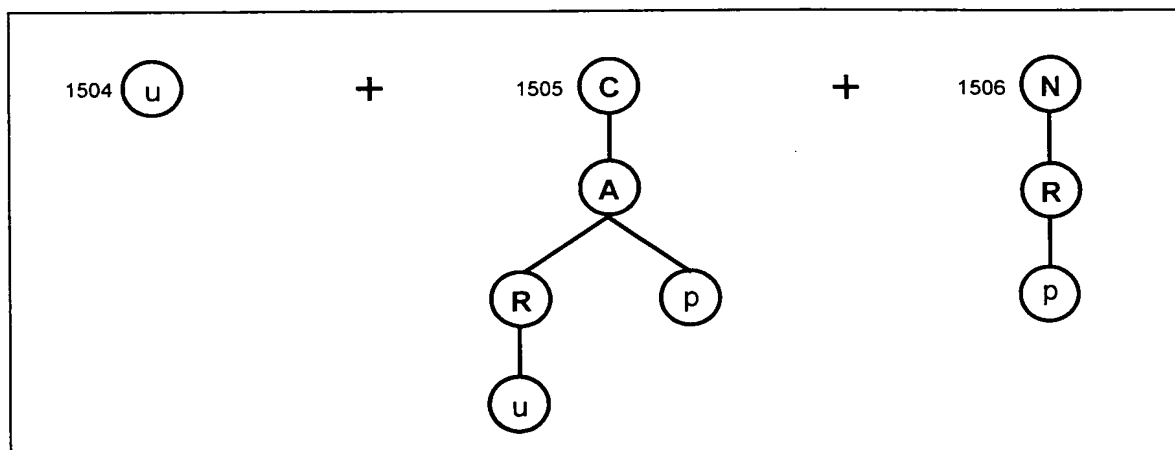
1502

Cromossoma

1503

uNpNttuptCARpuutupNRpCtutut

Árvore de expressão



**Figura 16**

Gavetas do estado inicial:

```
s#####-0  
vulnsiaer-1  
iuvr#####-2  
riev#####-3  
ui#####-4  
isunrl###-5  
uniav####-6  
lireav###-7  
ni#####-8  
#####-9
```

Cromossoma

1601

ARpCuututCARpuuuupACtptpuut

Árvore de expressão

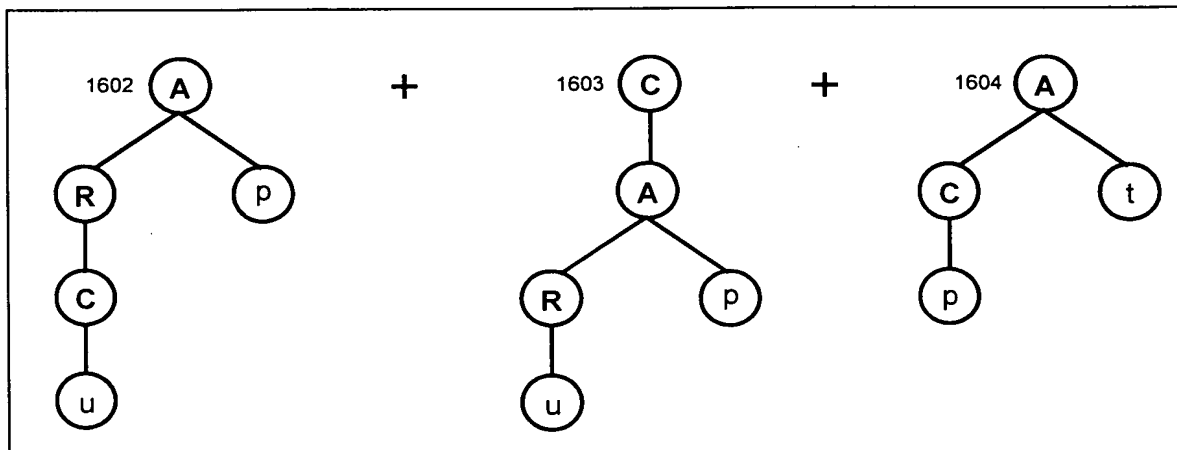


Figura 17

Gavetas do estado inicial:

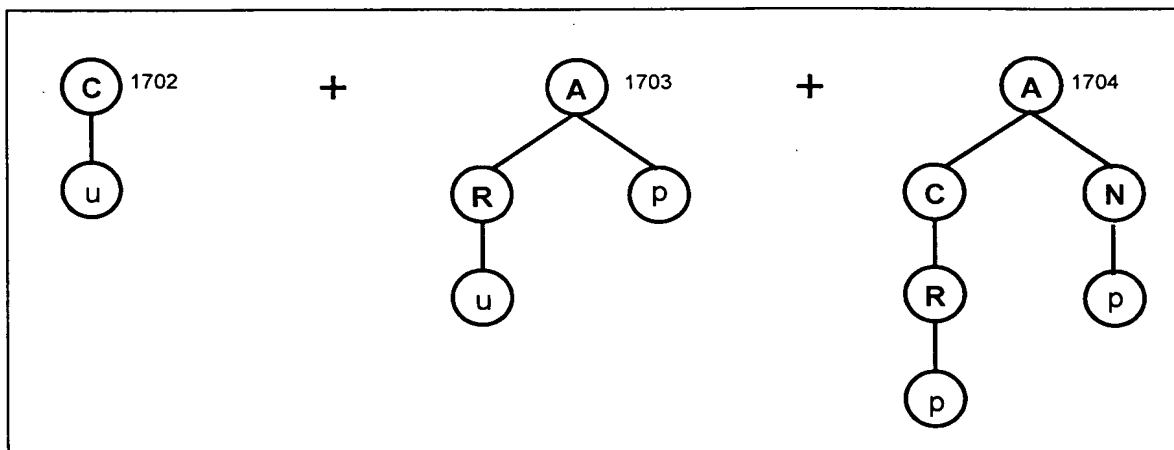
```
avnurle##-0
vsrui####-1
uerlvsnai-2
saelnu###-3
linv####-4
sivnrlaeu-5
vulrsaine-6
esla####-7
vnarlsei#-8
#####-9
```

Cromossoma

1701

CutputuptARpuuuttpACNRppuuu

Árvore de expressão



**Tabela 2**

	Invenção Presente	Programação Genética
G	100	51
P	30	500
C	10	167
$P_s$	0,70	0,767
$R_z$	4	4
$F_z$	120000	17034000

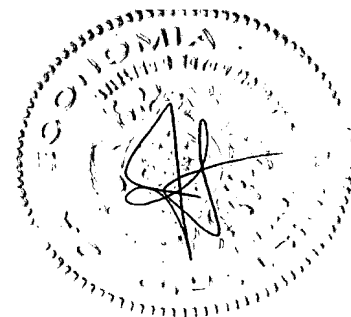
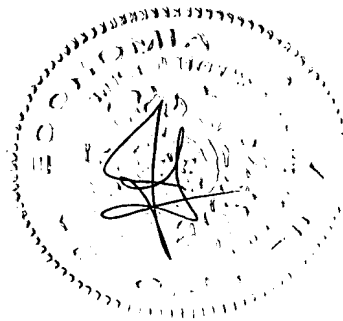


Figura 18



Cromossoma

1801

Aptidão

AO31N4322a4AAAbb342444AAAaN244bb3AAA2Nbb3a1b = 44

Árvore de expressão

1802

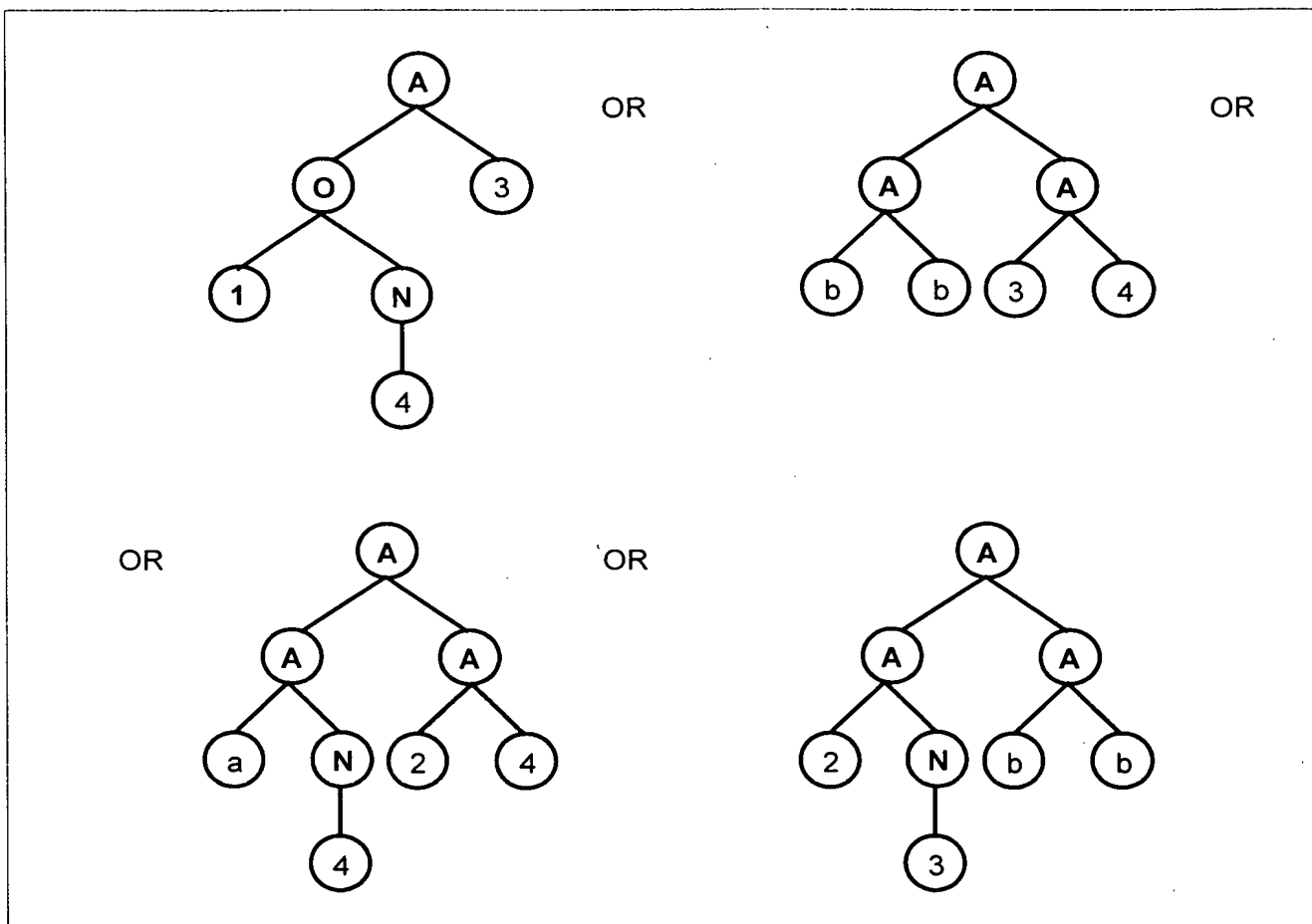


Figura 19

Cromossoma

1901

Aptidão

A1A2Ob4443aAa030133311Aa03bbbb322AO21Abb4a33 = 132

Árvore de expressão

1902

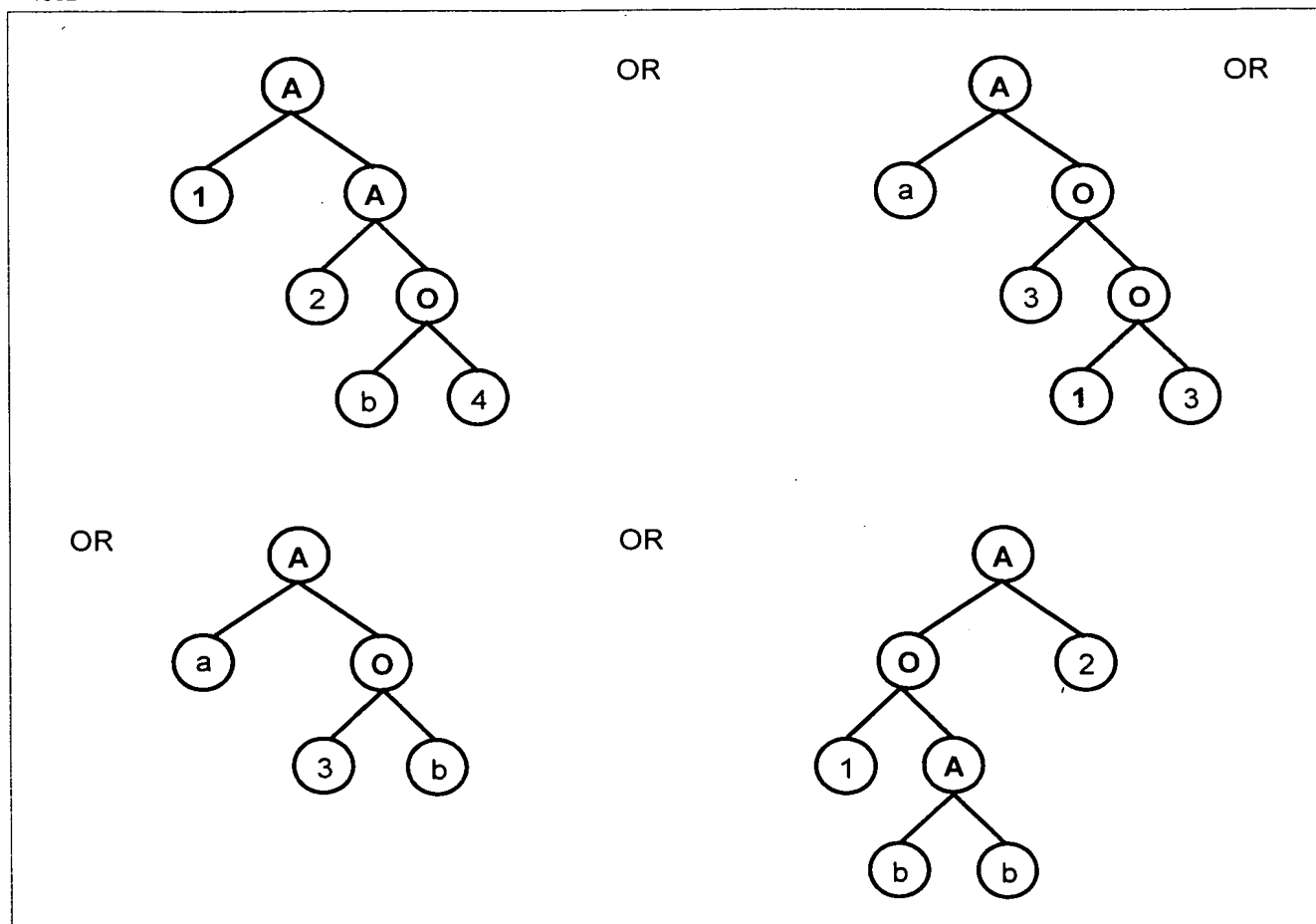


Figura 20

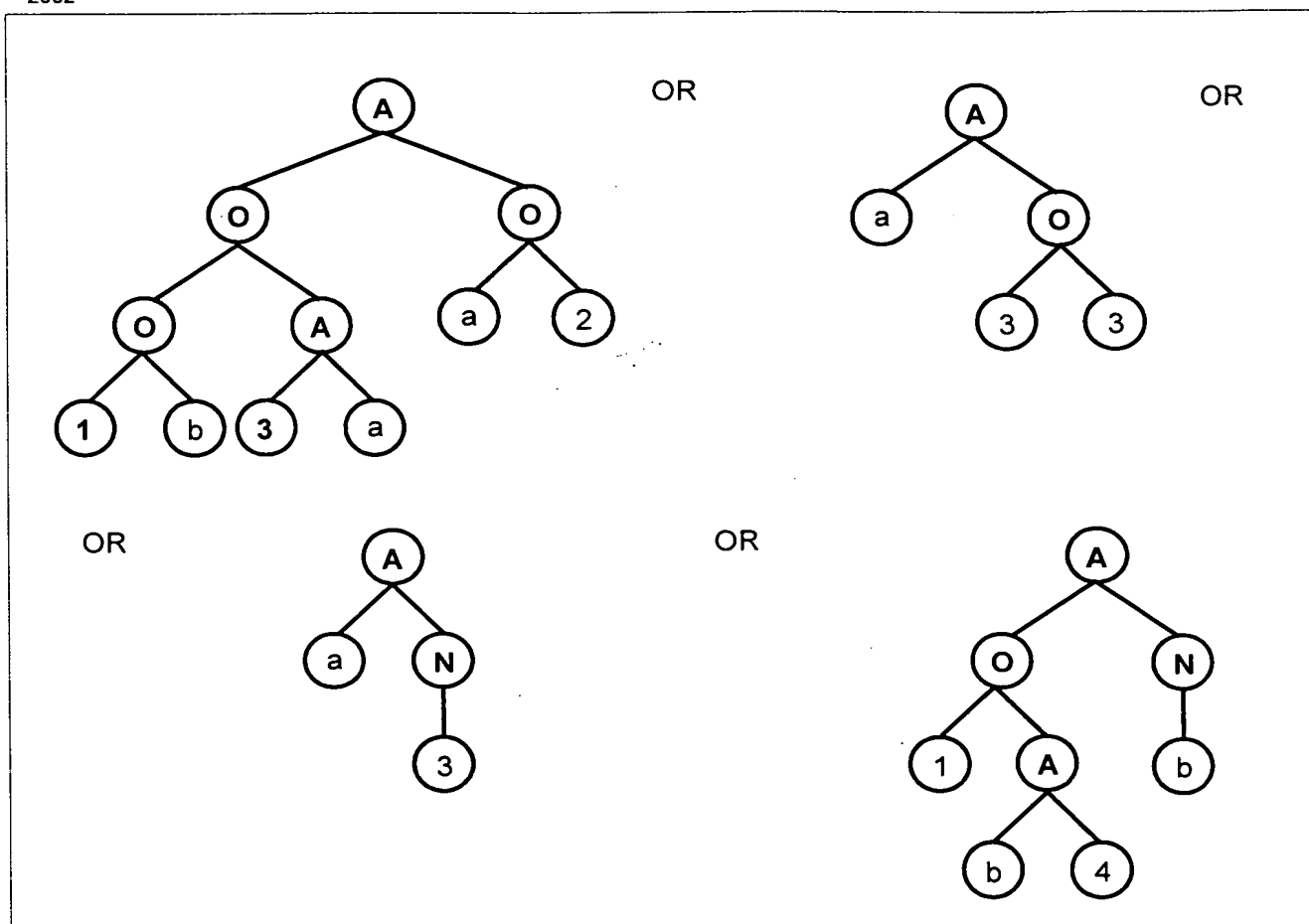


Cromossoma

2001 Aptidão  
A000Aa21b3aAa033133311AaN31bb4321AON1Abb4a3b = 216

Árvore de expressão

2002





**Figura 21**



Cromossoma

2101

---

AAN2baa4b2bAAO4Ab33a31AaANA3b4312AON1bb1b233 = 310

Aptidão

Árvore de expressão

2102

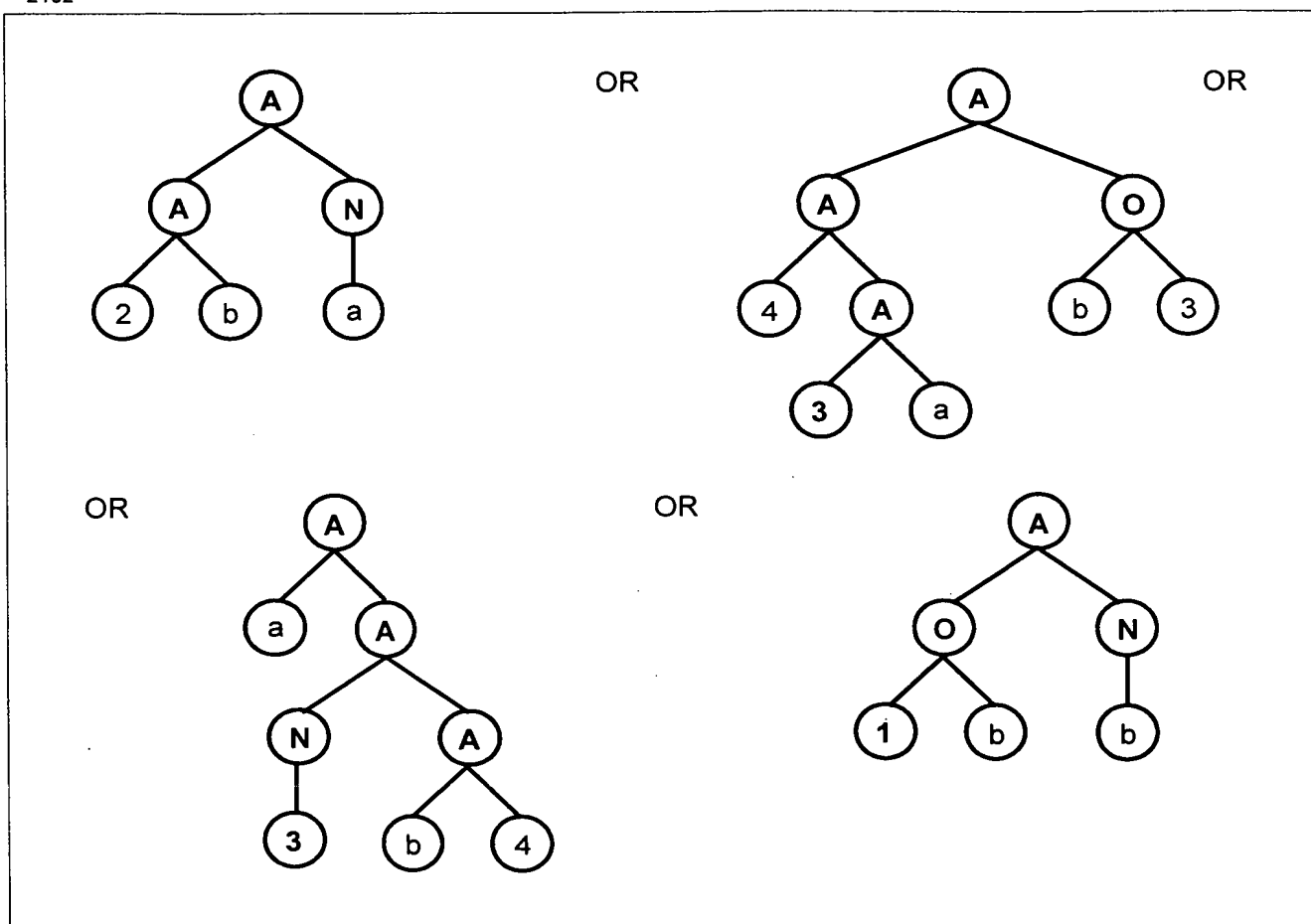


Figura 22

Cromossoma

2201 Aptidão  
 AANAb22a41AAObAb14a2bAaA3Nba1111AAN1Nba2a2b = 400

Árvore de expressão

2202

